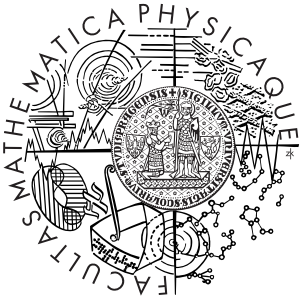
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Lenka Mišániková

Veřejná mince

Katedra algebry

Vedoucí bakalářské práce: Mgr. Štěpán Holub, Ph.D.
Studijní program: Obecná matematika

2008

Ďakujem Mgr. Štěpánu Holubovi, Ph.D. za vedenie tejto bakalárskej práce, jeho trpezlivosť a cenné rady.

Prehlasujem, že som svoju bakalársku prácu napísala samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce a jej zverejňovaním.

V Prahe dňa 4.8.2008 Lenka Mišániková

Obsah

Názov práce: Verejná minca
Autor: Lenka Mišániková
Katedra (ústav): Katedra algebry
Vedúci bakalárskej práce: Mgr. Štěpán Holub, Ph.D.
e-mail vedúceho: holub@karlin.mff.cuni.cz

Abstrakt: V tejto práci predstavíme interaktívne dôkazové systémy a Arthur-Merlinove hry, a ukážeme vzťahy medzi dvoma zložitostnými triedami, ktoré sú nimi definované. Oba systémy sú založené na komunikácii dvoch Turingových strojov, „svedka" a „sudcu" s využitím náhodnosti. V interaktívnych dôkazových systémoch je táto náhodnosť „známa" len „sudcovi", preto sa nazýva aj systém so súkromnou mincou. V Arthur-Merlinových hrách je náhodnosť známa svedkovi aj sudcovi - z čoho pochádza názov systém s verejnou mincou. Arthur-Merlinove hry sú triviálne špeciálnym prípadom interaktívnych dôkazových systémov. Goldwasser a Sipser[6] ukázali, že triedy definované interaktívnymi dôkazovými systémami a Arthur-Merlinovými hrami sú ekvivalentné z pohľadu rozpoznávania jazyka. Dôkaz je uvedený v siedmej kapitole tejto práci.

Kľúčové slová: interaktívne dôkazové systémy, Arthur-Melinove hry, súkromná minca, verejná minca

Title: Public Coin
Author: Lenka Mišániková
Department: Department of algebra
Supervisor: Mgr. Štěpán Holub, Ph.D.
Supervisor's e-mail address: holub@karlin.mff.cuni.cz

Abstract: In this thesis we study interactive proof systems and Arhur-Merlin games. We will discuss the complexity classes they determine. Both models consist of two communicating Turing machines, „prover" and „verifier". The verifier is probabilistic. In interactive proof systems, the verifier has a private source of random bits, while in Arthur-Merlin games, the source is public. Arthur-Merlin games is a trivial, special case of interactive proof system. Goldwasser and Sipser[6] proved, that these two systems are equivalent in power with respect to language recognition. The proof is showed in this work.

Keywords: interactive proof systems, Arthur-Merlin games, private coin, public coin

Kapitola 1

Úvod

> Čo intuitívne očakávame od procedúry dokazujúcej pravdivosť nejakej vety? Za prvé, že je možné „dokázať" pravdivé tvrdenie. Za druhé, že je nemožne „dokázať" nepravdivé tvrdenie. Za tretie, že komunikácia dôkazu je efektívna v nasledujúcom zmysle. Nie je dôležité, ako dlho bude dokazovateľ počítať, kým na dôkaz príde. Kľúčové je, aby výpočty, ktoré musí previesť overovateľ, boli ľahké.
>
> Goldwasser, Micali, Rackoff 1985

Cieľom práce je zrozumiteľným spôsobom definovať interaktívne dôkazové systémy a Arthur-Merlinove hry a dokázať, že zložitostné triedy, ktoré určujú, sú ekvivalentné z pohľadu rozpoznávania jazyka.

Interaktívne dôkazové systémy aj Arthur-Merlinove hry sú založené na komunikácii dvoch Turingových strojov, „svedka" a „sudcu" s využitím náhodnosti. V interaktívnych dôkazových systémoch je táto náhodnosť „známa" len „sudcovi". V Arthur-Merlinových hrách je náhodnosť známa svedkovi aj sudcovi.

V nasledujúcej kapitole si pripomenieme niektoré základne pojmy z teórie zložitosti. Tretia kapitola je venovaná interaktívnym dôkazovým systémom. Zadefinujeme interaktívny dôkazový systém a predstavíme triedy, ktoré sú ním určené. Uvedieme problém grafového neizomorfizmu a dokážeme, že existuje jednoduchý intaraktívny systém, ktorý ho rieši. V štvrtej kapitole zadefinujeme Arthur-Merlinove hry a triedy nimi určené. V ďalšej kapitole dokážeme jednoduchší vzťah medzi týmito dvoma systémami. Šiesta kapitola predstavuje Goldwasser-Sipser protokol a lema na asymptotický dolný odhad. V siedmej kapitole predstavíme dôkaz ekvivalencie medzi týmito dvoma systémami.

Kapitola 2

Triedy P, NP a BPP

V tejto kapitole si pripomenieme niektoré základné pojmy z teórie zložitosti. Zadefinujeme základné triedy **P**, **NP** a **BPP** a uvedieme vzťahy medzi nimi. Ak sa pripustí náhodnosť, je trieda **BPP** prirodzeným rozšírením triedy **P**. Pochopenie tohto jednoduchého rozšírenia je dôležité, keďže interaktívne dôkazové systémy sú podobným, avšak náročnejším, rozšírením triedy **NP**.

Množstvo problémov skúmaných teóriou zložitosti je rozhodovacích. Rozhodovací problém je problém, ktorý má booleovskú odpoveď áno/nie. Napr. problém, či je k prvočíslo, je rozhodovací. Každý rozhodovací problém generuje jazyk, ktorý pozostáva z tých prvkov - „slov", na ktoré je odpoveď kladná, tj. áno. V našom príklade teda patria do tohto jazyka („byť prvočíslo") všetky prvočísla. V tejto práci sa budeme zaoberať výsostne rozhodovacími problémami.

Turingov stroj má podobu na obe strany nekonečnej pásky, rozdelenej na políčka. Táto páska má hlavu, ktorá v danom okamihu ukazuje na niektoré políčko.

Definícia 1. Turingov stroj
Nech Σ a Q sú neprázdné konečné množiny. Množinu Σ nazveme množinou symbolov a množinu Q množinu stavov. Množina Σ obsahuje symbol $\square$, ktorý znamená prázdne políčko. Množina Q obsahuje jeden význačný prvok q_s, ktorý nazývame začiatočný stav. Nech $\mathcal{M} = \{\leftarrow, \rightarrow, -\}$.
Program Turingového stroja je ľubovoľná podmnožina $P \subseteq Q \times \Sigma \times Q \times \Sigma \times \mathcal{M}$. Ľubovoľný prvok $(q, s, q', s', m) \in P$ sa nazýva inštrukcia. Turingov stroj je štvorica $T = (Q, \Sigma, q_s, P)$.

Turingov stroj T spolu so svojím vstupom $w \in \Sigma^*$ definuje výpočet. Σ^* značí množinu konečných postupností symbolov z množiny Σ. Sprá-

vanie stroja je riadené programom, ktorý je množinou inštrukcií. Pod inštrukciou $(q, s, q', s', m) \in P$ chápeme, že čítaný symbol sa zmení z s na s', stav hlavy sa zmení z q na q' a hlava prevedie pohyb m. Okamžitý popis stroja je dvojica $(q, s) \in Q \times \Sigma$, kde q je momentálny stav hlavy a s je symbol na políčku, na ktoré hlava ukazuje.

V nejakom okamžitom popise môže mať Turingov stroj viac inštrukcií. Výpočet teda nie je definovaný jednoznačne. Turingove stroje považujeme vo všeobecnosti za nedeterministické. Špeciálnym prípadom sú deterministické Turingove stroje. Tie majú pre každý okamžitý popis práve jednu inštrukciu.

Definície **TIME** a **NTIME** neuvádzame, je možné nájsť ich napríklad v [7].

Definícia 2. (Trieda **P**)
Nech n je veľkosť vstupu. Potom

$$\mathbf{P} = \bigcup_{k \in N} \mathbf{TIME}(n^k).$$

V kontexte Turingových strojov je to množina jazykov, ktoré sú rozhodnuteľné deterministickým Turingovým strojom v polynomiálnom čase. Keďže algoritmy pracujúce v polynomiálnom čase sú všeobecne považované za použitelné v praxi, problémy patriace do **P** sú niekedy označované ako zvládnutelné.

Definícia 3. Hovoríme, že nedeterministický Turingov stroj prijíma vstup w práve vtedy, keď na vstupe w existuje aspoň jeden prijímací výpočet. Ak žiaden prijímací výpočet neexistuje, stroj vstup zamieta.

Definícia 4. (Trieda **NP**)
Trieda **NP** (Non-deterministic Polynomial time) je množina problémov rozhodnuteľných nedeterministickým Turingovým strojom v polynomiálnom čase. Píšeme

$$\mathbf{NP} = \bigcup_{k \in N} \mathbf{NTIME}(n^k).$$

Existuje ekvivalentná definícia triedy **NP**. Tá hovorí, že **NP** je množina takých jazykov, že existuje binárna relácia overiteľná v polynomiálnom čase $R(w, y)$ a konštanta $c \geq 1$ tak, že pre každý vstup w platí

$$w \in L \Leftrightarrow (\exists y)(|y| \leq |w|^c \text{ a } R(w, y)).$$

Neformálne sa dá povedať, že y je overenie a $R(w, y)$ je overovateľ náležania w v jazyku L. Triedu **P** môžeme teda chápať ako množinu

jazykov, pre ktoré existuje polynomiálny algoritmus. Triedu **NP** môžeme chápať ako množinu jazykov, pre ktoré existuje polynomiálny overovateľ.

Trieda **BPP**(Bounded error probability in polynomial time) je trieda problémov, ktoré sú riešiteľné pravdepodobnostným Turingovým strojom v polynomiálnom čase tak, že omyl (nesprávne posúdenie toho, či $x \in L$) nastane v menej ako jednej tretine prípadov. Ešte pripomeňme, že pravdepodobnostný Turingov stroj je nedeterministický Turingov stroj, ktorý sa rozhoduje pre nejaké pokračovanie s danou pravdepodobnosťou.

Definícia 5. (Trieda **BPP**)
Jazyk L je v triede **BPP** práve vtedy, keď existuje pravdepodobnostný Turingov stroj pracujúci v polynomiálnom čase tak, že

1. ak $x \in L$, tak pravdepodobnosť prijatia je aspoň $\frac{2}{3}$.

2. ak $x \notin L$, tak pravdepodobnosť odmietnutia je aspoň $\frac{2}{3}$.

Ak sa teda pripustí náhodnosť, je trieda **BPP** prirodzeným rozšírením triedy **P**. To znamená, že každý jazyk, ktorý je v **P**, je aj v **BPP**. Otvoreným problémom ostáva, či tieto dve triedy sú ekvivalentné. Tiež vieme, že platí

$$\mathbf{P} \subseteq \mathbf{NP}.$$

Je možné, že aj tieto dve triedy sú ekvivalentné, opäť je to otvorený problém.
Vzťah medzi **NP** a **BPP** je nejasný, zatiaľ nebola dokázaná ani jedna inkluzia.

Kapitola 3

Interaktívne dôkazové systémy

V tejto kapitole si podrobne rozoberieme interaktívne dôkazové systémy a predstavíme triedy, ktoré sú nimi určené. Oboznámime sa s problémom grafového neizomorfizmu a dokážeme, že jazyk pozostávajúci z dvojíc neizomorných grafov je rozhodnuteľný interaktívnym dôkazovým systémom.

3.1 Súkromná minca

Nápad študovať interaktívne dôkazy sa objavil s nástupom rozvoja kryptografie. Keď človek potreboval preukázať svoju totožnosť (napr. na internete), napísal heslo a odoslal ho na overenie. Tento postup však nebol veľmi bezpečný, keďže niekto mohol heslo zachytiť a následne zneužiť. Preto sa začalo rozmýšlať, ako by človek dokázal, že pozná heslo bez toho, aby ho naozaj napísal. Goldwasser, Micali a Rackoff [4] skúmali, čo to vlastne znamená „napísať dôkaz" a tak prišli s nápadom interaktívneho dôkazového systému, dúfajúc, že vystihnú najväčšiu možnú triedu efektívne rozhodnuteľných jazykov. Interaktívne dôkazové systémy zavádzajú dva nové atribúty do procesu dokazovania nejakého tvrdenia, a to interakciu a náhodnosť. Treba mať však na zreteli, že tento systém nevie dokázať náležanie v prísnom matematickom zmysle, pretože pripúšťa náhodnosť a tak môže chybne uznať nejaké trvdenie za (ne)pravdivé.

Majme tvrdenie T a majme dôkaz D. Chceme overiť, že D je naozaj dôkazom tvrdenia T. Majme „sudcu" V (verifier), ktorý posudzujeje správnosť dôkazu. Chceli by sme, aby náš systém mal dve vlastnosti a to úplnosť a korektnosť.

Definícia 6. (Úplnosť a korektnosť)

- úplnosť: Ak tvrdenie T platí, potom existuje taký dôkaz D, že ho sudca V prijme
- korektnosť: Ak tvrdenie T neplatí, potom sudca V zamietne každý dôkaz D

Zjednodušene povedané, úplnosť hovorí, že pravdivé tvrdenie sa dá dokázať. Korektnosť hovorí, že nepravdivé tvrdenie sa dokázať nedá.

Interaktívny dôkazový systém pozostáva z dvoch častí, sudcu V (verifier) a svedka P (prover), ktorí spolu komunikujú formou vymieňania si správ. Aby tento systém bol použitelný v praxi, počet správ, potrebných na prijatie/zamietnutie vstupu, ktoré si vymenia, je ohraničený polynómom, ktorého premennou je dĺžka vstupu. Svedok je „všemocný génius" s neobmedzenou výpočtovou silou. Cieľom svedka je presvedčiť sudcu o pravdivosti nejakého výroku. Sudca pracuje v polynomiálnom čase a môže využívať náhodnosť v tom zmysle, že si môže „hádzať mincu" a pýtať sa otázky v závislosti od výsledkov týchto hodení. Kľúčovým faktom je, že toto hodenie je „súkromné", to znamená, že svedok nevie, čo si sudca hodil. Odtiaľ aj pochádza názov dôkazy so súkromnou mincou. Sudca sa môže pýtať polynomiálne veľa krát v závislosti od dĺžky vstupu a na konci rozhodne, či ho svedok presvedčil alebo nie.

Definícia 7. Trieda **IP**
Interaktívny dôkazový systém pre jazyk L je protokol (V, P) medzi dvoma Turingovými strojmi P a V definovanými nasledovne:

$$\begin{aligned} V: &\quad \Sigma^* \times \Sigma^* \times \Sigma^* \to \Sigma^* \cup \{\text{prijíma, zamieta}\}. \\ P: &\quad \Sigma^* \to \Sigma^*. \end{aligned}$$

P sa nazýva svedok (prover) a V sa nazýva sudca (verifier). P je Turingov stroj s neobmedzenou výpočtovou silou a V je pravdepodobnostný Turingov stroj pracujúci v polynomiálnom čase v závislosti od dĺžky vstupu.

Označme s_i zreťazenie i párov správ, $s_i = \#x_1\#y_1\#\cdots\#x_i\#y_i$. Pod $V(w, r, s_i) = x_{i+1}$ chápeme, že V so vstupom w, náhodnou postupnosťou r a doterajšími správami s_i vyprodukuje ako najbližšiu správu x_{i+1}. Pod značením $P(s_i\#x_{i+1}) = y_{i+1}$ chápeme, že P na vstupe $s_i\#x_{i+1}$ dá výstup y_{i+1}. Výmena jedného páru správ sa nazýva kolo.

Pre daný vstup w a náhodnú postupnosť r povieme, že

$$(V, P)(w, r) \text{ prijíma}$$

ak existuje postupnosť správ $s = \#x_1\#y_1\#\cdots\#x_l\#y_l$ takých, že $V(w, r, s) =$ prijíma a pre každé $i < l$ platí $V(w, r, s_i) = x_{i+1}$ a $P(s_i\#x_{i+1}) = y_{i+1}$.

Pre jednoduchosť predpokladajme, že máme funkciu ℓ takú, že pre každý vstup w dĺžky n, V prijme, len ak dĺžka r je $\ell(n)$. Potom píšeme

$$\Pr\left[(V, P)(w) \text{ prijíma}\right]$$

na označenie toho, že $\Pr\left[(V, P)(w, r) \text{ prijíma}\right]$ pre r zvolené náhodne z $\Sigma^{l(|w|)}$. Pod $\Pr\left[(V)(w) \text{ prijíma}\right]$ chápeme $\max_P \Pr\left[(V, P)(w) \text{ prijíma}\right]$. Jazyk, ktorý rozhoduje nejaký sudca V, značíme

$$L(V) = \left\{ w : \Pr\left[V(w) \text{ prijíma}\right] > \frac{1}{2} \right\}.$$

Protokol (V, P) musí spĺňať aj úplnosť a korektnosť, avšak v mierne upravenej podobe. Keďže pripúšťame náhodnosť, riskujeme chybu. Úplnosť a korektnosť je pre interaktívne dôkazové systémy definovaná nasledovne.

- Úplnosť (pre prvky je ľahké dokázať náležanie)

$$w \in L \Rightarrow \exists P : \Pr\left[(V, P)(w) \text{ prijíma}\right] \geq c.$$

- Korektnosť (pre neprvky je ťažké dokázať náležanie)

$$w \notin L \Rightarrow \forall P : \Pr\left[(V, P)(w) \text{ prijíma}\right] \leq s.$$

Štandardne sa za s volí $\frac{1}{3}$. Ako si ukážeme neskôr v amplifikačnej leme, môžeme si za s zvoliť ľubovolné iné číslo ostro menšie ako $\frac{1}{2}$. Za c sa spravidla volí $1 - s$. Je preto možné písať e namiesto s a $1 - e$ namiesto c. Číslo e potom nazývame pravdepodobnosť omylu.

Ak $c = 1$, hovoríme, že dôkazový systém má perfektnú úplnosť. Ak $s = 0$ povieme, že dôkazový systém má perfektnú korektnosť.

Interaktívny dôkazový systém definuje hierarchiu jazykov. Jazyk L je v $\mathbf{IP}(k)$ ak existuje interaktívny dôkazový systém s k kolami. Nech n je dĺžka vstupu. Potom platí, že

$$\mathbf{IP}(poly) = \cup_{c \geq 1} \mathbf{IP}(n^c).$$

IP značí **IP**$(poly)$.

Uveďme si príklad. Predstavme si, že máme kamaráta, ktorý tvrdí, že vie rozoznať chuť Coca Coly od chute Pepsi Coly. My mu neveríme, a tak spravíme test. Kamarát sa otočí chrbtom, my nalejeme do pohára jeden z nápojov. Kamarát ho ochutná a má povedať, o ktorý nápoj ide. Ak povedzme 100-krát za sebou správne odpovie, uveríme mu. Ak nie, vieme, že nápoje nevie rozoznať. Samozrejme, nenalejeme do pohára vždy Pepsi colu, ale hodíme si mincu a podľa nej sa rozhodneme, ktorý nápoj nalejeme.

Ide o jednoduchý dôkazový systém. My sme sudca a kamarát je svedok. My potrebujeme náhodnosť, lebo inak by sme vždy naliali ten istý nápoj. Predpokladajme, že kamarát nevie rozoznať nápoje. Zaujíma nás pravdepodobnosť, že napriek tomu mu uveríme. Pri 100 pokusoch by si kamarát musel 100-krát správne tipnúť. Pravdepodobnosť tejto situácie však je $\left(\frac{1}{2}\right)^{100}$, čo je veľmi malé číslo. Pri rastúcom počte opakovaní testu ide dokonca táto pravdepodobnosť k nule. Čiže kamarátova šanca oklamať nás je prakticky nulová. Naopak, ak kamarát nápoje rozoznať vie, vždy uhádne, ktorý nápoj sme mu naliali. Situácia, keď kamarát nápoje vie rozoznať a naschvál povie nesprávnu možnosť nastať nemôže, pretože z definície hry sa kamarát snaží hru vyhrať, a to môže len vtedy, keď odpovie správne.

- **IP** $= \{L | L$ má interaktívny dôkazový systém$\}$.
- Ako sme spomenuli v predchádzajúcej kapitole, trieda **BPP** zavádza náhodnosť do triedy **P**. Interaktívne dôkazové systémy tiež zavádzajú náhodnosť, avšak akoby do triedy **NP**.
- Svedok nemusí byť pravdepodobnostný Turingov stroj. Keďže má neobmedzenú výpočtovú silu, vie si vypočítať hodnoty hodu mincou, ktoré maximalizujú pravdepodobnosť, že sudca prijme a prispôsobiť ďalšie výpočty týmto hodom.
- Naopak, sudca musí byť pravdepodobnostný Turingov stroj, inak dostaneme len triedu **NP**. Ak je totiž sudca deterministický, potom prijíma/zamieta vstup s pravdepodobnosťou práve 1. Vychádzajme z predchádzajúceho pozorovania (svedok je deterministický) a z toho, že $c > 0$. Dostaneme, že (V, P) vždy prijíma prvky L. Keďže sudca môže písať a čítať len polynomiálne veľa symbolov v závislosti od dĺžky vstupu w, dĺžka zápisu komunikácie (V, P) je polynomiálna v dĺžke vstupu, a tak je polynomiálne dlhým dôkazom toho,

že w patrí do L. Konkrétne, postupnosť odpovedí svedka presvedčí sudcu o náležaní w.

- Ak nepovolíme interakciu medzi sudcom a svedkom, dostaneme $\mathbf{IP} = \mathbf{BPP}$.
- Z predchádzajúcich dvoch pozorovaní máme $(\mathbf{NP} \cup \mathbf{BPP}) \subset \mathbf{IP}$.
- Všetky jazyky patriace do triedy **IP** su rozhodnuteľné interaktívnym dôkazovým systémom s perfektnou úplnosťou. Toto je netriviálny fakt. Pre podrobnejšie spracovanie témy vrátane dôkazu, viď [3].
- Každý jazyk rozhodnuteľný interaktívnym dôkazovým systémom s perfektnou korektnosťou nutne patrí do triedy **NP**.

3.2 Grafový neizomorfizmus a IP

Definícia 8. Dva grafy $G = (V_G, E_G)$ a $H = (V_H, E_H)$ sú izomorfné, ak existuje bijekcia π medzi vrcholmi týchto grafov

$$\pi : V_G \to V_H$$

taká, že pre každé dva vrcholy $u, v \in V_G$ platí, že

$$(u, v) \in E_G \Leftrightarrow (\pi(u), \pi(v)) \in E_H. \tag{3.1}$$

Dva grafy G a H sú izomorfné, ak jeden vznikne ako permutácia druhého, ide vlastne o „premenovanie vrcholov" grafu G.

Definícia 9.
Jazyk ISO $= \{(G, H)$ také, že G je izomorfný s $H\}$.
Jazyk NONISO $= \{(G, H)$ také, že G nie je izomorfný s $H\}$.

Ak sú dva grafy izomorfné, tak nedeterministický Turingov stroj dokáže nájsť bijekciu, ktorá spĺňa podmienku 3.1. Overiť túto podmienku pre danú bijekciu je možné v polynomiálnom čase, a preto ISO $\in$ **NP**. Overiť dôkaz, že dva grafy nie sú izomorfné, už nie je také jednoduché, pretože zatiaľ neexistuje dôkaz, ktorý by bol overiteľný v polynomiálnom čase. Zovšeobecnene, hoci vieme rozpoznávať jazyk kladných odpovedí (napr.ISO) pre nejaký problém nedeterministickým Turingovým strojom v polynomiálnom čase, neznamená to, že takýmto strojom vieme

v tomto čase rozpoznávať aj jazyk záporných odpovedí (napr. NONISO). Fakt, že $L \in \mathbf{NP}$ nutne neznamená, že $\text{coL} \in \mathbf{NP}$, kde $\text{coL} = \Sigma^* - L$.

Nevie sa, či NONISO patrí do **NP**. Taktiež sa nevie, či patrí do **BPP**. Goldreich, Micali a Wigderson [5] ukázali, že na tento problém existuje jednoduchý interaktivny dôkazový systém a teda NONISO $\in$ **IP**.

Veta 10. NONISO $\in$ **IP**.

Dôkaz. Myšlienka je veľmi podobná, ako v príklade, kde sa nás kamarát snažil presvedčiť, že rozozná Pepsi Colu od Coca Coly. Povedzme, že sú nám (sudcovi) dané dva grafy G a H. My ich za chrbtom „zamiešame" - zpermutujeme. Vyberieme si náhodne jednu permutáciu a spýtame sa svedka, ktorého grafu je táto permutácia. Ak nie sú izomorfné, svedok by nemal mať problém správne nám odpovedať.

Teraz popíšeme protokol, ktorým svedok presvedčí sudcu, že grafy G a H nie sú izomorfné. Budeme predpokladať, že oba grafy majú rovnaký počet vrcholov - n (inak sú triviálne neizomorfné).

1. Sudca - zvolí si náhodne bit $b \in \{0, 1\}$, minca je spravodlivá, pravdepodobnosť každej voľby je $\frac{1}{2}$. Táto voľba určí s ktorým grafom bude sudca ďalej pracovať.

2. Sudca - zvolí náhodnú permutáciu π vrcholov $1, 2, \ldots, n$. Ak $b = 0$, pošle svedkovi $\pi(G)$, ak $b = 1$, pošle $\pi(H)$.

3. Svedok - povie ktorý graf mu bol poslaný (G alebo H). Ak G a H nie sú izomorfné, svedok vždy vráti správnu odpoveď. Ak sú izomorfné, svedok nevie posúdiť, či permutácia vznikla z grafu G alebo H a odpovie náhodne. Pravdepodobnsť, že odpovie správne je $\frac{1}{2}$.

Nato, aby bola splnená podmienka korektnosti, protokol musí prebehnúť aspoň dva krát. Dá sa to však spraviť v jednom kole. Sudca si na začiatku zvolí náhodne dva bity. V závislosti na každom bite zvlášť si zvolí permutáciu, dostane dve nezávislé permutácie. Tie pošle svedkovi, ktorý musí každej priradiť správny graf. Ak svedok odpovedal vždy správne, sudca prijme. Inak zamietne. Opakovanou aplikáciou protokolu pravdepodobnosť chyby klesá.

Vráťme sa k definícii interaktívnych dôkazových systémov (def. 7), konkrétne k požiadavkám úplnosti a korektnosti.

Ak G a H nie sú izomorfné, existuje svedok P tak, že

$$\Pr[(V, P) \text{ prijíma}] = 1.$$

Ak G a H sú izomorfné, pri počte opakovaní k,

$$\Pr[(V, P) \text{ prijíma}] \leq \left(\frac{1}{2}\right)^k.$$

Ak sú grafy izomorfné, svedok ich nevie rozlíšiť a v kroku 3 si náhodne (s pravdepodobnosťou $\frac{1}{2}$) zvolí G alebo H.

□

Kapitola 4

Arthur-Merlinove hry a verejná minca

> Triedu **IP** môžeme jednoducho modifikovať. Stačí, keď sudca zverejní výsledky svojho hádzania mincou. Tým dostaneme novú triedu, takzvané Arthur-Merlinove hry. Problematikou Arthur-Merlinových hier sa budeme zaoberať v tejto kapitole.

Kráľ Arthur vie, že Merlin má nadprirodzené schopnosti, ale nedôveruje mu. Ako ho teda má Merlin, všemocný čarodejník, presvedčiť, že slovo w patrí do jazyka L? Arthur sa môže pýtať ľubovolné otázky, Merlin odpovedá tak, aby presvedčil Arthura, že pozná odpoveď. Na základe týchto odpovedí Arthur nakoniec usúdi, či ho Merlin presvedčil alebo nie(teda kto vyhral). Počas hry Arthur môže hádzať mincou, hra pripúšťa náhodnosť. Avšak výsledok hodu musí vždy povedať Merlinovi (z toho aj pojem „verejná minca"). Arthur, ako smrteľník, je obmedzený polynomiálnym časom, ktorý závisí na dĺžke vstupu. Komunikácia musí tiež prebehnúť v polynomiálnom čase.

Arthur je teda polynomiálny pravdepodobnostný stroj, rovnako ako sudca v **IP**. Merlin má rovnako ako svedok v **IP**, neobmedzenú výpočtovú moc a vždy volí optimálnu odpoveď. Arthur-Merlinove hry sú skoro totožné s triedou **IP**, jediný rozdiel je v tom, že namiesto súkromnej mince sa používa verejná. S konceptom Arthur-Merlinových hier prišiel v roku 1985 László Babai [1].

Definícia 11. Arthur-Merlinove hry
Arthur Merlinove hry sú definované ako trieda **IP** (viď definícia 7) s nasledujúcim rozdielom. Za náhodný vstup r sa považuje zreťazenie l reťazcov (l je počet kôl), $r = r_1 r_2 \cdots r_l$. Sudca V musí vyprodukovať r_i

ako svoju i-tú správu. Teda pre $i \leq l$, $V(w, r, s_i) = r_i$ alebo prijíma alebo zamieta.

Definícia 12. Označme dĺžku vstupu $|w| = n$. Nech $t(n)$ je polynomiálna funkcia v premennej n. Jazyky prijímané Arthur-Merlinovými hrami v i kolách $i \leq t(n)$, tvoria triedy $\mathbf{AM}(t(n))$ (keď prvý ťahá Arthur) a $\mathbf{MA}(t(n))$ (prvý ťahá Merlin). Ďalej platí

$$\mathbf{AM}(poly) = \mathbf{MA}(poly) = \bigcup_{k>0} \mathbf{AM}(n^k).$$

Ďalej definujeme $\mathbf{AM}(1) = \mathbf{A}$, $\mathbf{AM}(2) = \mathbf{AM}$, $\mathbf{AM}(3) = \mathbf{AMA}$, $\mathbf{MA}(2) = \mathbf{MA}$ a analogicky.

Triedy, v ktorých posledný ťahá Merlin (napr. $\mathbf{AM}(2)$, či $\mathbf{MA}(3)$) vyvolávajú otázku. Problém je v tom, že Arthur má podľa definície vo svojom poslednom ťahu prijať alebo zamietnuť vstup. To však nemá ako spraviť, keďže posledný ťahá Merlin. V odbornej literatúre je to riešené rôzne. Najčastejšie sa predpokladá, že po poslednom(Merlinovom) ťahu sa opäť dostáva na ťah Arhtur, avšak už iba rozhoduje, či vstup prijme alebo zamietne. Inými slovami, už nemôže použiť náhodnosť. V práci [2] je predstavený rozhodca. Ten na konci každej Arthur-Merlinovej hry rozhodne o tom, kto vyhral.

Pozorovanie 13. *Ak Arthur ignoruje Merlina a sám rozhodne problém, dostaneme triedu* $\mathbf{BPP}$. *Platí teda*

$$\mathbf{AM}(1) = \mathbf{BPP}.$$

Pretože Merlin je nedeterministický máme

$$\mathbf{MA}(1) = \mathbf{NP}.$$

V roku 1985 Babai a Moran [2] ukázali, že pre každú konštantu $k > 1$

$$\mathbf{AM}(2) = \mathbf{AM}(k) = \mathbf{MA}(k + 1).$$

Tiež dokázali, že pre $t(n)$ polynomiálne ohraničené platí

$$\mathbf{AM}\left(2t(n)\right) = \mathbf{AM}\left(t(n)\right).$$

Otázkou však ostáva, či môže táto redukcia prebehnúť bez toho, aby sa zvýšila zložitosť Merlina. Keďže však má Merlin neobmedzenú výpočtovu silu, redukcia je korektná.

Kapitola 5

$\mathbf{IP}(poly) \subseteq \mathbf{AM}(poly)$

> Zaujímavou otázkou je, v akom vzťahu sú interaktívne dôkazové systémy a Arthur-Merlinove hry. Je zrejmé, že Arthur-Merlinove hry sú podtriedou interaktívnych dôkazových systémov. Špeciálne platí $\mathbf{AM}(poly) \subseteq \mathbf{IP}(poly)$. V tejto kapitole dokážeme, že aj $\mathbf{IP}(poly) \subseteq \mathbf{AM}(poly)$ a teda $\mathbf{IP}(poly) = \mathbf{AM}(poly)$

Pripomeňme si, že interaktívne dôkazové systémy sa líšia od Arthur-Merlinových hier zdrojom náhodnosti. Nasledujúca veta hovorí, že každý protokol so súkromnou mincou môže byť nahradený protokolom s verejnou mincou a perfektnou korektnosťou pre protokoly s polynomiálnym počtom kôl.

Veta 14. $\mathbf{IP}(poly) \subseteq \mathbf{AM}(poly)$

Dôkaz. Chceme ukázať, že ak jazyk L patrí do triedy $\mathbf{IP}(poly)$, tak patrí aj do triedy $\mathbf{AM}(poly)$. Pre overovanie $w \in L$ máme teda už korektný protokol so súkromnou mincou. Cieľom dôkazu je na základe tohto protokolu konštruovať nový protokol, ktorý bude používať verejnú mincu. Následne musíme ukázať, že tento nový protokol spĺňa podminku úplnosti a korektnosti.

Uvažujme protokol so súkromnou mincou, ktorý nám overuje náležanie prvku $w \in L$. Majme v tomto protokole sudcu V a svedka P, ktorý je optimálny (to znamená, že v protokole **IP** maximalizuje šancu prijatia). Z definície protokolu **IP** vyplýva, že

$$w \in L \Rightarrow \Pr[(V,P)(w) \text{ prijma}] > \frac{2}{3},$$

$$w \notin L \Rightarrow \Pr[(V,P)(w) \text{ prijma}] < \frac{1}{3}.$$

Skonštruujeme teraz protokol z **AM**. V tomto protokole majme sudcu „Arthura" V' a svedka „Merlina" P'. Predstavme si binárny strom reprezentujúci možné komunikácie medzi V a P v protokole **IP**. Každá cesta z koreňa tohto stromu do niektorého z listov reprezentuje jeden náhodný reťazec vybraný sudcom V, čo predstavuje jednu možnú komunikáciu. Listy stromu (koncové vrcholy) označíme „prijíma"/„zamieta" podľa toho, či komunikácia zodpvedajúca ceste z koreňa do tohto listu prijma alebo odmieta w. To znamená, že ak $w \in L$, aspoň $\frac{2}{3}$ listov budú označené „prijať". Zároveň keď $w \notin L$, tak maximálne $\frac{1}{3}$ listov bude označená „prijať". To vyplýva z vlastnosti protokolu patriaceho do triedy **IP**.

Definujme si teraz komunikáciu medzi P' a V', o ktorej neskôr ukážeme, že spĺňa požiadavky kladené na protokol triedy **AM**. Máme vstup w. Svedok P' pošle počet listov $N'(start)$, označených ako „prijíma" v strome reprezentujúcom možné komunikácie medzi V a P pre vstup w. Tento počet si Arthur uloží do koreňa stromu. Arthur bude teraz postupne prechádzať z koreňa stromu do jedného z listov nasledovne: V každom vrchole (vrátane koreňa) sa spýta pre každé z dvoch detí tohto vrchola na počty listov $N'(0)$ a $N'(1)$, označených „prijma" v podstrome, ktorého koreňom je dané dieťa. Merlin mu pošle tieto údaje a následne Arthur skontroluje, či platí

$$N' = N'(0) + N'(1).$$

Ak vzťah neplatí, Arthur zamietne vstup w. Ak platí, Arthur si údaje uloží do vrcholov a s pravdepodobnosťou $\frac{N'(i)}{N'}$ pokračuje vo vrchole $N'(i)$. Takto rekurzívne postupuje až príde do listu.

Teraz ukážeme, že popisovaný protokol spĺňa podmienku úplnosti a korektnosti.

- úplnosť: Nech $w \in L$. Potom pravdepodobnosť, že vyberieme list s hodnotou „zamieta" je nulová. Vieme, že aspoň $\frac{2}{3}$ listov má hodnotu prijíma. Stačí teda, aby svedok vždy udal pravdivú hodnotu prijímacích listov. To aj spraví, viď definícia svedka. Dostávame protokol s perfektnou úplnosťou a platí

$$\text{ak } w \in L \Rightarrow \Pr[V'(w) \text{ prijíma}] = 1.$$

- korektnosť: Nech $N(\sigma)$ označuje skutočný počet listov s hodnotou prijíma v podstrome s koreňom σ a $N'(\sigma)$ počet týchto listov, ktorý udal Merlin. Ak Merlin udal, že ich je viac ako v skutočnosti (teda $N'(\sigma) > N(\sigma)$), zistíme to s pravdepodobnosťou aspoň $1 - \frac{N(\sigma)}{N'(\sigma)}$. Dokážeme to indukciou. Ak je σ listom, sme hotoví, pretože $N(\sigma) =$

0 alebo 1. Nech tvrdenie platí pre deti $N(\sigma_0)$ a $N(\sigma_1)$ každého vrcholu $N(\sigma)$. Pravdepodobnosť odhalenia Merlinovho klamstva je rovná minimálne

$$\begin{aligned}\left(1-\frac{N(\sigma_0)}{N'(\sigma_0)}\right)\frac{N'(\sigma_0)}{N'(\sigma)}+\left(1-\frac{N(\sigma_1)}{N'(\sigma_1)}\right)\frac{N'(\sigma_1)}{N'(\sigma)} &= \\ =\frac{1}{N'(\sigma)}[N'(\sigma_0)+N'(\sigma_1)-N(\sigma_0)-N(\sigma_1)] &= \\ =\frac{N'(\sigma)-N(\sigma)}{N'(\sigma)} &= =1-\frac{N(\sigma)}{N'(\sigma)}.\end{aligned}$$

Keďže $N'(\sigma)>\frac{2}{3}$ a $N(\sigma)<\frac{1}{3}$, vidíme, že

$$x\notin L\Rightarrow \Pr[V' \text{ prijíma}]<\frac{1}{2}.$$

Celý protokol spustíme 2 krát. Dostávame

$$x\notin L\Rightarrow \Pr[V' \text{ prijíma}]<\frac{1}{4}.$$

Všimnime si, že protokol medzi Arthurom a Merlinom je naozaj s verejnou mincou, keďže v každom kroku Arthur pošle Merlinovi údaj, v ktorom dieťati vcholu sa rozhodol pokračovať.
Tiež môže nastať otázka, prečo neskúsime prejsť z koreňa stromu do každého listu a tým si jednoznačne overiť, či Arthurovi hovorí Merlin pravdu o počte listov s hodnotou „prijma". Avšak uvedomme si, že to by sme museli prejsť 2^n ciest, kde n je počet hodov mince sudcu z triedy **IP**. To by sme sa však dostali mimo polynomiálny čas. □

Kapitola 6

Asymptotický dolný odhad

V tejto kapitole predstavíme Goldwasser-Sipser protokol a lema na asymptotický dolný odhad.

V dôkaze, ktorý predstavíme v nasledujúcej kapitole, Merlin bude chcieť Arthurovi dokázať, že množina náhodných reťazcov, ktoré ho presvedčia prijať vstup, je „veľká". Preto potrebujeme protokol, ktorým Merlin dokáže presvedčiť Arthura, či je nejaká množina naozaj „veľká".

Predstavme si, že máme množinu $S \subseteq \{0,1\}^k$, ktorej prvky vie Arthur rozoznať v polynomiálnom čase a funkciu $N(k)$. Arthur má vstup prijať, ak $|S| \geq N(k)$ a zamietnuť, napr. ak $|S| \leq \frac{N(k)}{10k^2}$. Chceme nájsť Arthur-Merlinov protokol na rozhodnutie tohto problému. Uvedomme si, že pre $N(k)$ polynómialne, je to triviálne: ak je $|S| \geq N(k)$, Merlin pošle ľubovolných $N(k)$ prvkov z S. Ak ich aspoň $N(k)$(správnych) nepošle, Arthur zamietne.

Nech $N(k) = 2^k$. To znamená, že Arthur prijíma, ak každý prvok $\{0,1\}^k$ je v S. Naopak, Arthur zamieta, ak je prvok v množine S s pravdepodobnosťou najviac $\frac{1}{10k^2}$. Zostrojiť Arthur-Merlinov protokol (dokonca s perfektnou úplnosťou), nie je ťažké. Arthur vyberie náhodne $x \in \{0,1\}^k$, a prijme práve vtedy, keď $x \in S$. Ak $N(k) = 2^k$, Arhtur vždy prijme. Ak $N(k) \neq 2^k$, Arthur (nesprávne) prijme s pravdepodobnosťou najviac $\frac{1}{10k^2}$.

Nech $N(k)$ je naďalej veľká, ale $N(k) \neq 2^k$. Povedzme, $N(k) = \frac{2^k}{100}$. Merlin na začiatku pošle Arthurovi $O(k)$ reťazcov $x_i \in \{0,1\}^k$. Následne si Arthur náhodne vyberie jedno $x \in \{0,1\}^k$ a pošle ho Merlinovi. Merlin chce ukázať, že pre nejaké i platí, že $x \oplus x_i \in S$. Ak je S „veľká", existuje množina $\{x_1, \cdots, x_{O(k)}\}$, ktorej každý prvok „posunieme" o rovnakú konštantu a aspoň jeden „posunutý" prvok bude v S. Tento typ protokolu

prestane fungovať, akonáhle je S dostatočne malé na to, aby Merlin našiel množinu, ktorej posunutie má neprázdny prienik s S. Toto nastáva napríklad pre $N = 2^{\sqrt{k}}$, čo je netriviálny výsledok.

V najvšeobecnejšom prípade(a pre nás najzaujímavejšom), nastáva situácia, že S je oveľa menšie ako $\{0,1\}^k$. Na túto situáciu použijeme hashovanie. Zmyslom hashovania je zobraziť veľkú množinu (v našom prípade $\{0,1\}^k$) na menšiu množinu, povedzme $\{0,1\}^b$, kde $b \ll k$. Chceme, aby väčšina prvkov $\{0,1\}^b$ mala približne $2^{-k}|S|$ obrazov v S (čo je očakávaný počet, ak by bola h úplne náhodna funkcia). Konkrétne, ak má S veľkosť 2^b, očakávame, že zobrazenie bude „skoro bijekcia" S na $\{0,1\}^b$ a teda $h(S)$ bude „veľká" podmnožina. Dostávame obdobu predchádzajúceho prípadu. Merlin pošle l prvkov $y_i \in \{0,1\}^b$. Arthur vyberie $y \in \{0,1\}^b$. Následne Merlin pošle i a $x \in \{0,1\}^k$ tak, že $h(x) = y \oplus y_i$, pre $x \in S$. Ide o korektný Arthur-Merlinov protokol. Dôkaz vynecháme.

Goldwasser-Sipser protokol

1. Sudca V zvolí l náhodných funkcií $f_1, \cdots, f_l : \Sigma^k \to \Sigma^b$ a l^2 reťazcov $z_1, \ldots, z_{l^2} \in \Sigma^b$.
2. Sudca pošle svedkovi $f_1, \cdots, f_l, z_1, \ldots, z_{l^2}$.
3. Svedok pošle sudcovi x.
4. Sudca prijme ak $x \in S$ a $f_i(x) = z_j$ pre nejaké i, j, kde $1 \leq i \leq l$ a $1 \leq j \leq l^2$.

Definícia 15. Nech D je $k \times b$ booleovská matica. Lineárna funkcia $h_D : \Sigma^k \to \Sigma^b$ je daná predpisom $h_D(x) = x.D$. Používame klasické násobenie maticou modulo 2. Náhodným výberom matice D získame náhodnu lineárnu funkciu. Ak $H = \{h_1, h_2, \ldots, h_l\}$ je množina funkcií, $C \subseteq \Sigma^k$, $D \subseteq \Sigma^b$ potom $H(C)$ značí $\bigcup h_i(C)$ a $H^{-1}(D)$ značí $\bigcup h_i^{-1}(D)$. Nech $|C|$ je veľkosť množiny C.

Nasledujúce lema pravdepodobnoste popisuje úspešnoť Goldwasser-Sipser protokolu s daným $b, k, l > 0$ a $l > max\{b, 8\}$.

Lema 16. *Lema na asymptotický dolný odhad množiny*
Nech $b, k, l > 0$, $l > max(b, 8)$ a $C \subseteq \Sigma^k$. Náhodne vyberieme l lineárnych funkcií $H = \{h_1, \ldots, h_l\}$, $h_i : \Sigma^k \to \Sigma^b$ a l^2 reťazcov $Z = \{z_1, \ldots, z_{l^2}\} \subseteq \Sigma^b$. Potom platia nasledujúce tvrdenia.

1. *Ak $b = 2 + \lceil log\,|C| \rceil$, potom platí*

a)

$$\Pr[|H(C)| \geq \frac{|C|}{l}] \geq 1 - 2^{-l}, \tag{6.1}$$

b)

$$\Pr[H(C) \bigcap Z \neq \emptyset] \geq 1 - 2^{-l/8}.$$

2. a)

$$|H(C)| \geq l\,|C|\,.$$

b) Nech $d > 0$ a $|C| \leq 2^b/d$. Máme

$$\Pr[H(C) \bigcap Z \neq \emptyset] \leq l^3/d.$$

Dôkaz.

1a) Vieme, že

$$b = 2 + \lceil log\,|C| \rceil.$$

Úpravou dostávame

$$\begin{aligned} 2^b &= 2^{2+\lceil log|C| \rceil}, \\ 2^b &= 4 \cdot 2^{\lceil log|C| \rceil}, \\ 2^b &\geq 4\,|C|\,. \end{aligned}$$

Nech $(h_i(x))^j$ označuje j-tý bit reťazca $h_i(x)$. Zvoľme $x, y \in \Sigma^k$, $x \neq y$, $i, j > 0$ pevné vždy keď nie je kvantifikované.
Pravdepodobnosť že dva bity sa rovnajú, je

$$\Pr[(h_i(x))^j = (h_i(y))^j] = \frac{1}{2}.$$

Reťazce $h_i(x)$ i $h_i(y)$ majú b bitov, preto pravdepodobnosť, že sa tieto reťazce rovanjú je

$$\Pr[h_i(x) = h_i(y)] = \frac{1}{2^b}.$$

Vypočítame, aká je pravdepodobnosť, že pre ľubovolné slovo $x \in \Sigma^k$ nájdeme v množine C aspoň 1 také y, že pre aspoň jedno i platí, že h_i zobrazí x aj y na to isté slovo (teda $h_i(x) = h_i(y)$).

$$\begin{aligned} \Pr\left[(\exists y \in C)\,(x \neq y \wedge h_i(x) = h_i(y))\right] &\leq \frac{|C|}{2^{-b}} \leq \frac{1}{4}, \\ \Pr[(\forall i \leq l)(\exists y \in C)\,(x \neq y \wedge h_i(x) = h_i(y))] &\leq \frac{1}{4^l}, \\ \Pr\left[(\exists x \in C)(\forall i \leq l)\,(\exists y \in C)(x \neq y \wedge h_i(x) = h_i(y))\right] &\leq \frac{|C|}{4^l} \leq \frac{1}{2^l}. \end{aligned}$$

Podmienku

$$(\exists x \in C)(\forall i \leq l)(\exists y \in C)(x \neq y \wedge h_i(x) = h_i(y))$$

z poslednej nerovnosti označme ako tvrdenie q.
Dokazovanú nerovnosť 6.1 môžeme ekvivalentne napísať ako

$$\Pr[|H(C)| < \frac{|C|}{l}] \leq 2^{-l}.$$

Výraz zo zátvorky $|H(C)| < \frac{|C|}{l}$ označme ako tvrdenie p. Vieme, že platí

$$(p \Rightarrow q \wedge \Pr(q) \leq 2^{-l}) \Rightarrow \Pr(p) \leq 2^{-l}.$$

Preto nám stačí dokázať, že $p \Rightarrow q$, čo dokážeme nepriamo. Negáciou výroku q dostávame $\neg q$:

$$(\forall x \in C)(\exists i \leq l)(\forall y \in C)\,(x = y \vee h_i(x) \neq h_i(y))\,.$$

Predpokladajme teda, že $\neg q$ platí. Dostávame

$$|C| \leq \sum_{i \leq l} |h_i(C)| \leq \sum_{i \leq l} |H(C)| = |H(C)| \sum_{i \leq l} 1 = l \cdot |H(C)|\,,$$

kde druhá nerovnosť plynie z toho, že

$$|H(C)| = \bigcup_{i \leq l} h_i(C) \Rightarrow (\forall i)(|h_i(C)| \leq |H(C)|).$$

Máme teda

$$\frac{|C|}{l} \leq |H(C)|\,,$$

čo je vlastne $\neg p$ a sme hotoví.

1b) Najprv si uvedomme, že $\left|\sum^b\right| = 2^b$. Z toho, že

$$b = 2 + \lceil log\,|C| \rceil$$

dostaneme

$$\begin{aligned} 2^b &= 2^{2+\lceil log|C| \rceil}, \\ 2^b &= 4 \cdot 2^{\lceil log|C| \rceil} \leq 4 \cdot 2^{log|C|+1} = 8 \cdot |C|\,, \\ |C| &\geq \frac{2^b}{8}. \end{aligned}$$

Predpokladajme, že

$$|H(C)| \geq \frac{|C|}{l}.$$

Pomocou vyššie uvedených vzťahov dostávame

$$|H(C)| \geq \frac{2^b}{8l},$$
$$\frac{|H(C)|}{\left|\sum^b\right|} \geq \frac{1}{8l}.$$

Idea je nasledujúca. Približne každý „$8l$-tý" náhodný reťazec dĺžky b patrí do množiny $H(C)$. Vieme, že l je ostro väčšie ako 8 a Z obsahuje l^2 náhodných reťazcov. Predstavujeme si teda, že pravdepodobnosť, že aspoň jeden náhodný reťazec zo Z bude patriť aj do $H(C)$ je veľmi veľká. Presne,

$$\Pr[H(C) \bigcap Z = \emptyset] \leq (1 - \frac{1}{8l})^{l^2} + \frac{1}{2^l} < 2^{\frac{-l}{8}}.$$

Keby sme chceli byť presnejší, výraz $1 - \frac{1}{8l}$ by sme násobili pravdepodobnosťou, že $|H(C)| \geq \frac{|C|}{l}$. Táto pravdepodobnosť je podľa časti $1a$ rovná $1 - 2^{-l}$, takže jej hodnotu pri danom odhade zhora môžeme naozaj zanedbať. Výraz 2^{-l} v predchádzajúcej nerovnosti je pravdepodobnosť, že $|H(C)| < \frac{|C|}{l}$. Keby sme chceli byť presní, opäť by sme mali vynásobiť hodnotu 2^{-l} pravdepodobnosťou, že za podmienky $|H(C)| < \frac{|C|}{l}$, je $H(C) \bigcap Z = \emptyset$. Vieme, že pravdepodobnosť je nanajvýš rovná jednej a tak je horný odhad korektný. Ostáva nám dokázať poslednú nerovnosť

$$(1 - \frac{1}{8l})^{l^2} + \frac{1}{2^l} < 2^{\frac{-l}{8}}.$$

Platí, že

$$(1 - \frac{1}{8l})^{l^2} = e^{l^2 . \log(1 - \frac{1}{8l})} \leq e^{l^2 . (\frac{1}{-8l})} = e^{\frac{-l}{8}}.$$

Ak ukážeme, že

$$e^{\frac{-l}{8}} + 2^{-l} < 2^{\frac{-l}{8}}$$

budeme hotoví. To je ekvivalentné

$$\frac{2^l + e^{\frac{l}{8}}}{e^{\frac{l}{8}} 2^l} < 2^{\frac{-l}{8}},$$
$$2^l + e^{\frac{l}{8}} < 2^{\frac{7l}{8}} e^{\frac{l}{8}} = (2^{\frac{7}{8}} e^{\frac{1}{8}})^l.$$

Keďže $l > 8$, uvedené vzťahy naozaj platia.

2a) Stačí si uvedomiť, čo nerovnosť hovorí.

2b) Z 2a a predpokladu ľahkou úpravou dostávame

$$\frac{|H(C)|}{\left|\sum^b\right|} \leq \frac{l\,|C|}{d\,|C|} = \frac{l}{d}.$$

Pravdepodobnosť, že ľubovolné $z_i \in Z$ bude patriť do $H(C)$, je zhora obmedzená $\frac{l}{d}$. Keďže Z obsajuhe l^2 reťazcov, pravdepodobnosť, že aspoň jeden z nich bude patriť aj do $H(C)$ je $l^2.\frac{l}{d} = \frac{l^3}{d}$, čo je dokazovaná nerovnosť.

□

Kapitola 7

Verejná minca = súkromná minca

V tejto kapitole dokážeme, že pre každý polynóm $Q(n)$ platí, že $\mathbf{IP}(Q(n)) \subseteq \mathbf{AM}(Q(n)+2)$. Spolu s opačnou inkluziu, ktorá je triviálna a s tým, že pre každú konštantu $k > 1$ platí, že $\mathbf{AM}(k) = \mathbf{AM}(k+1)$ dostávame plnú ekvivalneciu týchto dvoch tried, teda $\mathbf{IP}(Q(n)) = \mathbf{AM}(Q(n))$.

Veta 17. *Pre každý polynóm $Q(n)$ platí:* $\mathbf{IP}(Q(n)) = \mathbf{AM}(Q(n)+2)$.

Najprv sa zamyslime, ako funguje $\mathbf{IP}(1)$ protokol.

V: náhodne si zvolí náhodný reťazec r. Na základe vstupu w a reťazca r sa sudca V rozhodne pre nejakú otázku x a pošle ju svedkovi P

P: vygeneruje odpoveď y_x bez toho aby poznal náhodný reťazec r a pošle y_x sudcovi

V: $V(w, r, x\#y_x)$ rozhodne, či prijma alebo zamieta

Podčiarknime, že otázka x nesmie prezradiť náhodný reťazec r, lebo by sme dostali verejnú mincu.

To, či sudca V prijme alebo zamietne je plne určené náhodným reťazcom r, ktroý si zvolil. Idea je tá, že keď $w \in L$, väčšina náhodných reťazcov „spôsobí" prijatie vstupu. Naopak, keď $w \notin L$ počet reťazcov, vďaka ktorým V prijme, je malý.

Nech V má exponenciálne malú pravdepodobnosť chyby e, posiela správy dĺžky m a používa náhodne reťazce dĺžky ℓ. Pre každé $x \in \Sigma^m$, nech $\beta_x = \{r : V(w, r, \#) = x\}$. Pre každé $y \in \Sigma^m$, nech $\alpha_{xy} = \{r : r \in \beta_x \wedge V(w, r, \#x\#y) = \text{prijma}\}$. Ideálny svedok sa bude snažiť

pre každé x zvoliť také y_x, ktoré maximalizuje $|\alpha_{xy}|$. Nech $\alpha_x = \alpha_{xy_x}$. Nech $\alpha_0 = \cup_x \alpha_x$. Potom $\Pr[V(w)\text{prijma}] = |\alpha_0|/2^l$. Cieľom Merlina je presvedčiť Arthura, že $|\alpha_0| > e \cdot 2^l$.

Zamerajme sa na simuláciu **IP**(1) protokolu. Zvážme jednoduchý prípad. Ak $x \in L$, nech pre všetky „otázky" x existuje „odpoveď" y taká, že $|\alpha_{xy}| \geq N$. Ak $x \notin L$, nech nech pre všetky x a pre všetky y platí, že $|\alpha_{xy}| \leq \frac{N}{10n^2}$. Naviac predpokladajme, že svedok pozná(a verí) hodnotu N. Môžeme použiť jednoduchý protokol. Najprv pošle P hodnotu x a y. Potom použije Goldwasser-Sipser protokol na overenie $|\alpha_{xy}| \geq \frac{2}{3}N$. Inými slovami, Merlin povie Arthurovi „tu je možný pár - otázka, odpoveď, ktorý by ťa prinútil prijať vstup w. Dokážem ti, že je pravdepodobné, že by si sa opýtal túto otázku". Avšak v obecnom prípade hodnota N závisí na otázke x (značíme N_x) a sudca nevie jej hodnotu.

Svedok teda spočíta „typickú" hodnotu N_x a nazve ju N. Potom dokáže, že pre „veľa" otázok platí $N_x \geq N$. Nato opäť použije Goldwasser-Sipser protokol.

Celý protokol vyzerá nasledovne. Merlin pošle Arthurovi hodnotu N. Pomocou Goldwasser-Sipser protokolu ukáže, že hodnota N platí pre „veľa" otázok. Ďalej si zvolí otázku x_0 a znova pomocou Goldwasser-Sipser protokolu ukáže, že $N_{x_0} \geq N$. Tým dokáže, že táto množina je naozaj „veľká".

Pre protokol s g kolami sa prvý Goldwasser-Sipser protokol iteruje na získanie hodnoty $\alpha_0 \supseteq \alpha_1 \supseteq \cdots \supseteq \alpha_g$, kde je „veľa" možností, ako zväčšiť α_{i-1} na α_i a α_g je „veľká".

Dôkaz. Dôkaz vety 17.

Nech $W \in IP[Q(n)]$. Bez ujmy na obecnosti môžeme predpokladať, že na vsupte w dĺžky n je práve $g(n) = Q(n)/2$ párov správ(teda kôl) poslaných medzi V a P. Tieto správy sú dlhé práve $m(n)$. Náhodny reťazec r je dlhý $\ell(n)$. Nech pravdepodobnosť chyby je $e(n)$.

Lema 18. *Amplifikačné lema*
Nech $p(n)$ je polynóm. Nech V je sudca, ktorý pracuje na vstupe dlhom n, s počtom správ najviac $g(n)$, každej dlhej $m(n)$, používa $\ell(n)$ náhodných bitov a má pravdepodobnosť chyby $\frac{1}{3}$. Potom existuje sudca V' taký, že $L(V) = L(V')$, pracujúci s $g(n)$ správami, každej dlhej $O(p(n)m(n))$, používajúc $O(p(n)l(n))$ náhodných bitov a s pravdepodobnosťou omylu zhora ohraničenou $2^{-p(n)}$.

Poďľa predchádzajúceho lema môžeme predpokladať, že

$$e(n) \leq \ell(n)^{-12g^2(n)}.$$

Ďalej môžeme predpokladať, že $\ell(n) > \max(g(n), m(n), 80)$. Píšeme g, m, e, ℓ namiesto $g(n)$, $m(n)$, $e(n)$ a $\ell(n)$.

Máme funkcie V a P. My popíšeme funkcie A(Arthur) a M(Merlin), ktoré budú simulovať V a P. Premenná x_i značí správy poslané A a y_i značí správy poslané M.
V prvom kroku popíšeme protokol medzi A a M. Najprv popíšeme protokol pre Arthura a potom pre Merlina. Dokážeme, že ide o korektný protokol Arthur-Merlinovej hry(spľňa úplnosť a korektnosť), ktorý simuluje (V, P) protokol.

Protokol pre Arthura

Kolo 0**.**
A dostane číslo b_1 od M. Pokračujeme kolom 1.

Kolo i $(1 \leq i \leq g)$**:**
Zatiaľ dostal A čísla $b_1, \ldots, b_i$ a náhodné reťazce $x_1, \ldots, x_{i-1}, y_1, \ldots, y_{i-1}$ od M. A si náhodne zvolí l lineárnych funkcií $H = \{h_1, \ldots, h_l\}$, $h_i : \Sigma^m \to \Sigma^{b+1}$ a l^2 reťazcov $Z = \{z_1, \ldots, z_{l^2}\} \subseteq \Sigma^{b+1}$ a pošle ich M. A dostane od M reťazce x_i a y_i a číslo b_i. A overí, či $x_i \in H^{-1}(Z)$. Ak to neplatí, A okamžite zamietne. Ak platí, A pokračuje kolom $i + 1$.

Posledné kolo $g + 1$**:**
Nech $s_i = x_1 \# y_1 \# \cdot \# x_i \# y_i$. A náhodne vyberie l lineárnych funkcií $H = \{h_1, \ldots, h_l\}$, $h_i : \Sigma^l \to \Sigma^{b_{g+1}}$ a l^2 reťazcov $Z \subseteq \Sigma^{b_{g+1}}$. A čaká, že následne mu M pošle reťazec $r \in \Sigma^l$. A overí, či $r \in H^{-1}(Z)$. A prijme, ak pre každé $i \leq g$ platí, že $V(w, r, s_i) = x_{i+1}$, $V(w, r, s_g) =$ prijma a $\sum b_i \geq l - g \log l$.

Môže Merlin presvedčiť Arthura?
Ukážeme, že $\Pr[V(w) \text{ prijíma}] > e(n) \Leftrightarrow \Pr[A(w) \text{ prijíma}] > \frac{2}{3}$.

($\Rightarrow$) **Merlinov protokol keď** $w \in L$.
Najrpv si zaveďme označenie. Pre $r \in \Sigma^l$ a pre prúd správ $s = v_1 \# v_2 \# \cdots \# v_k$ povieme, že

$$(V, P)(w, r) \text{ prijíma cez s,}$$

ak prvých k správ medzi V a P sa zhoduje so s a $(V, P)(w, r)$ prijíma.

Nech $\Pr[V(w) \text{ prijíma}] \geq \frac{2}{3}$. Zvoľme P ľubovolné také, že

$$\Pr[(V,P)(w) \text{ prijíma}] \geq \frac{2}{3}.$$

Pre M zostrojíme taký protokol, že $\Pr[(A,M)(w) \text{ prijíma}] \geq \frac{2}{3}$.

Kolo 0**:**
Nech $i = 1$. Pokračuj so „získaj b_1".

Získaj $b_i (i \leq g)$**:**
Nech $s_{i-1} = x_1 \# y_1 \# \cdots \# x_{i-1} \# y_{i-1}$ je prúd správ pre (V,P) protokol, ktoré boli zatiaľ vymenené. Nech pre každé $x \in \Sigma^m$ je $\alpha_x = \{r : (V,P)(w,r) \text{ prijíma cez } s_{i-1}\#x\}$. Vytvor triedy $\gamma_1, \ldots, \gamma_l$ tak, že γ_d obsahuje všetky α_x také, že $2^{d-1} < |\alpha_x| \leq 2^d$. Zvoľ γ_{max}, ktorej zjednotenie

$$\bigcup \gamma_{max} = \bigcup \{\alpha_x : \alpha_x \in \gamma_{max}\} \tag{7.1}$$

je najväčšie. Pošli $b_i = 2 + \lceil \log |\gamma_{max}| \rceil$.

Kolo i**:**
M dostane $h_1, \ldots, h_l$ a reťazce $z_1, \ldots, z_{l^2}$ od A. Ak existuje $x \in H^{-1}(Z)$ také, že $\alpha_x \in \gamma_{max}$, nazvi ho x_i. Následne M odpovedá dvojicou x_i, y_i kde $y_i = P(s_{i-1}\#x_i)$. Ak M nenájde takéto x, odpovedá „chybou". Ďalej značíme množinu α_{x_i} ako α_i. Priradíme $i \leftarrow i+1$. Pokračuj „získaj b_i".

Získaj b_{g+1}**:**
M vypočíta hodnotu b_{g+1} nasledovne. Nech $s_g = s_{g-1}\#x_g\#y_g$ je zvolený prúd správ. Máme $\alpha_g = \{r : (V,P)(w,r) \text{ prijíma cez } s_g\}$. Pošli $b_{g+1} = 2 + \lceil \log |\alpha_g| \rceil$.

Kolo $g+1$**:**
M dostane $h_1, \ldots, h_l$ a reťazce $z_1, \ldots, z_{l^2} \in \Sigma^{b_{g+1}}$ od A. Ak existuje $r \in \alpha_g \cap H^{-1}(Z)$, M pošle toto r. Inak M odpovie „chybou". (Všimnime si, že $r \in \alpha_g$ implikuje $V(w,r,s_g) =$ prijíma)
Koniec protokolu.

Teraz ukážeme, že $\Pr[(A,M)(w) \text{ prijíma}] \geq \frac{2}{3}$.
Nech $\alpha_0 = \{r : (V,P)(w,r) = \text{ prijíma}\}$.
Keďže $\Pr[V(w) \text{ prijíma}] \geq \frac{2}{3}$, tak $|\alpha_0| \geq (2/3)\, 2^l$.
S definície M vieme, že A akceptuje, ak M nikdy neodpovedá „chybou" a

$\sum b_i \geq l - g \log l$. Podľa lemy [16] vieme, že pravdepodobnosť, že M odpovie „chybou" v ľubovoľnom kole, je menšia nanajvýš rovná $2^{-l/8}$. Teda pravdepodobnosť, že M nikdy neodpovie „chybou", je zhora ohraničná $(g+2)2^{-l/8} \ll 1/3$

Nasledujú dve tvrdenia, ktoré dokazujú, že $\sum b_i \geq l - g \log l$.

Tvrdenie 19. *Pre každé* $0 \leq i < g$, *platí*

$$|\alpha_i| \geq \frac{|\alpha_{i-1}|}{l 2^{b_i}}.$$

Dôkaz. Zoberme si i-té kolo a množiny α_x definované v „získaj b_i". Z definície máme, že $\bigcup_x \alpha_x = \alpha_{i-1}$. Nezabudnime, že značíme $\alpha_{x_{i-1}}$ ako α_{i-1}. S odvolaním sa na vzťah 7.1, dostávame

$$\left|\bigcup \gamma_{max}\right| \geq \frac{|\alpha_{i-1}|}{l}.$$

Vieme, že jednotlivé členy množiny γ_{max} sa navzájom líšia najviac o násobok 2. Keďže $\alpha_i \in \gamma_{max}$, máme

$$|\alpha_i| \geq \frac{|\bigcup \gamma_{max}|}{2|\gamma_{max}|}.$$

Keďže $b_i = 2 + \lceil \log |\gamma_{max}| \rceil$, (viď „získaj b_i"), máme

$$2^{b_i} \geq 2|\gamma_{max}|.$$

Dostávame vzťah

$$|\alpha_i| \geq \frac{|\bigcup \gamma_{max}|}{2^{b_i}} \geq \frac{|\alpha_{i-1}|}{l 2^{b_i}}.$$

□

Tvrdenie 20.

$$\sum b_i \geq l - g \log l.$$

Dôkaz. Z predchádzajúceho tvrdenia 19 máme

$$|\alpha_g| \geq \frac{|\alpha_0|}{l^g \prod_{i \leq g} 2^{b_i}}.$$

Vieme, že $|\alpha_0| \geq (2/3)2^l$. Po zlogaritmovaní dostávame

$$\begin{aligned}
\log|\alpha_g| &\geq \log\left((2/3)2^l\right) - \log l^g \prod_{i\leq g} 2^{b_i} \geq \\
&\geq \log 2/3 + l - \left(g\log l + \sum_{i\leq g} \log 2^{b_i}\right) \geq \\
&\geq (l-1) - \left(g\log l + \sum_{i\leq g} b_i\right).
\end{aligned}$$

Vieme, že $b_{g+1} > 1 + \log|\alpha_g|$ (tak je zvolené b_{g+1} v „získaj b_{g+1})", a tak dostávame

$$\sum_{i\leq g+1} b_i \geq l - g\log l$$

a sme hotoví. □

($\Leftarrow$) Ak $w \notin L$, Merlin s „veľkou" pravdepodobnosťou nepresvedčí Arthura

Ukážeme, že keď $\Pr[V(w) \text{ prijíma}] \leq e$, potom $\Pr[A(w) \text{ prijíma}] \leq 1/3$.

Pre každé $i > 0$ a $s_i = x_1\#y_1\#\cdots\#x_i\#y_i$ označme

$$a(s_i) = \max_P \Pr[(V,P)(w) \text{ prijíma cez } s_i].$$

Nech pre každé $x \in \Sigma^m$, y_x označuje také $y \in \Sigma^m$, ktoré maximalizuje $a(s_i\#x\#y)$.

Neformálne, nasledujúce tri výroky maju za úlohu ukázať, že $a(s_{i+1})$ je s veľkou pravdepodobnosťou oveľa menšie ako $a(s_i)$.

Tvrdenie 21.

$$a(s_i) = \sum_x a(s_i\#x\#y).$$

Zvoľme $0 \leq i < g$ a s_i. Pre každé $c > 0$, označme

$$X_c = \{x : \ a(s_i\#x\#y_x) \geq a(s_i)/c\}.$$

Tvrdenie 22.

$$|X_c| \leq c.$$

Dôkaz. Vďaka predchádzajúcemu tvrdeniu, vieme, že

$$a(s_i) \geq \sum_{x \in X_c} a(s_i \# x \# y_x).$$

Z definície množiny X_c dostávame

$$\begin{aligned} a(s_i) &\geq \sum_{x \in X_c} \frac{a(s_i)}{c} = \frac{a(s_i)}{c}|X_c|, \\ c &\geq |X_c| \end{aligned}$$

□

Zvoľme $b, d > 0$. Vyberme l náhodných lineárnych funkcií $H = \{h_1, \ldots, h_l\}$, $h_i : \Sigma^m \to \Sigma^b$ a l^2 náhodných reťazcov $Z \subseteq \Sigma^b$. Vyberme ľubovolné $x \in H^{-1}(Z)$ a ľubovolné $y \in \Sigma^m$. Nech $s_{i+1} = s_i \# x \# y$.

Nasleduje popis udalostí, ktoré maximalizujú pravdepodobnosť, že Merlin presvedčí Arthura o tom, že $w \in L$ napriek tomu, že to tak nie je. Pomenujme nasledujúcu udalosť E_{i+1}:

$$a(s_{i+1}) \geq \frac{a(s_i)}{2^b/d}.$$

Tvrdenie 23.

$$\Pr[E_i] \leq l^3/d.$$

Dôkaz. Nech $c = \lfloor 2^b/d \rfloor$. Potom $|X_c| \leq 2^b/d$ (viď tvrdenie 22). Z definície y_x máme $a(s_i \# x \# y_x) \geq a(s_{i+1})$. Dostávame, že ak $a(s_{i+1}) \geq a(s_i)/(2^b/d)$, potom $x \in X_c$. Keďže $x \in H^{-1}(Z)$, máme

$$\begin{aligned} \Pr\left[a(s_{i+1}) \geq \frac{a(s_i)}{2^b/d}\right] &= \Pr\left[x \in X_c \cap H^{-1}(Z)\right] = \\ &= \Pr\left[H(X_c) \cap Z \neq 0\right] \leq \\ &\leq l^3/d. \end{aligned}$$

Posledná nerovnosť vyplýva z lema na dolný odhad, časť $2b$. □

Zvoľme s_g. Vyberme l náhodných lineárnych funkcií $H = \{h_1, \ldots h_l\}$, $h_i : \Sigma^l \to \Sigma^{b_{g+1}}$ a l^2 náhodných reťazcov $Z \subseteq \Sigma^{b_{g+1}}$. Vyberme ľubovolné $r \in H^{-1}(Z)$. Nazvime nasledujúcu udalosť E_{g+1}.

$$2^l a(s_g) \leq 2^b/d \wedge (V, P)(w, r) \text{ prijíma cez } s_g.$$

Tvrdenie 24.

$$\Pr[E_{g+1}] \leq l^3/d.$$

Dôkaz. Z definície $a(s_i)$ máme

$$\frac{|\{r : (V,P)(w,r) \text{ prijíma cez } s_g\}|}{2^l} = a(s_g).$$

Malou úpravou dostávame

$$|\{r : (V,P)(w,r) \text{ prijíma cez } s_g\}| = 2^l a(s_g).$$

Ak nastane E_{g+1}, budú splnené predpoklady lema na dolný odhad časť $2b$, ktoré použijeme a sme hotoví. □

V kole i (teda $b = b_i$) nastane udalosť E_i s pravdepodobnosťou najviac l^3/d. Keďže kôl je $g+1$, nastane udalosť E_i aspoň v jednmom kole s pravdepodobnosťou najviac $(g+1)l^3/d$. Preto zvolíme

$$d = 3(g+1)l^3.$$

Dostávame, že $\Pr[(\exists i) \text{ nastane } E_i] \leq 1/3$.

Nech E_i nenastane. Ukážeme, že A zamietne ak predpokladáme, že

$$\Pr[V(w) \text{ prijíma}] \leq e.$$

Predpokladáme, že platí $(\forall i \leq e)(\neg E_i)$. Máme

$$a(s_{i+1}) \leq \frac{a(s_i)}{2^{b_i}/d}.$$

Dostávame

$$a(s_g) \leq \frac{a(s_0)}{\prod_{i \leq g} 2^{b_i}/d}. \tag{7.2}$$

Pretože $\neg E_{g+1}$, musí platiť buď

$$(V,P)(w,r) \neq \text{ prijíma},$$

alebo

$$2^l a(s_g) \geq 2^{b_{g+1}}/d. \tag{7.3}$$

Ak $(V,P)(w,r)$ prijíma, musí platiť vzťah 7.3. Kombináciou tohto vzťahu a vzťahu 7.2 dostávame

$$2^l a(s_0) \geq \prod_{1 \leq i \leq g+1} (2^{b_i}/d).$$

Po zlogaritmovaní máme

$$\begin{aligned}
\log\left(2^l a(s_0)\right) &\geq \log\left(\prod_{1\leq i\leq g+1}(2^{b_i}/d)\right),\\
\log 2^l + \log a(s_0) &\geq \sum_{1\leq i\leq g+1}\log\frac{2^{b_i}}{d},\\
l + \log a(s_0) &\geq \sum_{1\leq i\leq g+1}\left(\log 2^{b_i} - \log d\right) =\\
&= \sum_{1\leq i\leq g+1} b_i - (g+1)\log d =\\
&= \sum_{1\leq i\leq g+1} b_i - (g+1)\log\left(3(g+1)l^3\right).
\end{aligned}$$

Keďže $l \geq g+1$, dostávame

$$\begin{aligned}
l + \log a(s_0) &\geq \sum_{1\leq i\leq g+1} b_i - (g+1)\log\left(3l^4\right) \geq\\
&\geq \sum_{1\leq i\leq g+1} b_i - (g+1)\left(\log 3 + 4\log l\right) \geq\\
&\geq \sum_{1\leq i\leq g+1} b_i - 2g\ .\ 5\log l =\\
&= \sum_{1\leq i\leq g+1} b_i - 10g\log l.
\end{aligned}$$

Vieme, že

$$a(s_0) = \Pr\left[V(w) \text{ príme}\right] \leq e \leq l^{-12g^2}.$$

Preto dostávame,

$$\begin{aligned}
l - 12g^2\log l &\geq \sum_{1\leq i\leq g+1} b_i - 10g\log l,\\
\sum_{1\leq i\leq g+1} b_i &\leq l - (12g-10)\,g\log l < l - g\log l.
\end{aligned}$$

Spomeňme si, že Arthur prijme len vtedy, keď $(V,P)(w,r)$ prijíma a $\sum b_i \geq l - g\log l$. Ak pre žiadne $i \leq g+1$ nenastane E_i a $\Pr\left[V(w) \text{ prijíma}\right] \leq e$, Arthur vždy zamieta. Keďže $\Pr\left[\exists i :\ \text{nastane } E_i\right] \leq 1/3$, platí, že

$$\Pr\left[A(w) \text{ prijíma}\right] \leq 1/3.$$

□

Kapitola 8

Záver

V úvodnej kapitole sme si pripomenuli niektoré základné triedy zložitosti. Následne sme si predstavili triedu interaktívnych dôkazových systémov **IP** a Arthur-Merlinových hier **AM**. Defínicie týchto tried sme si podrobnejšie rozobrali. Načrtli sme problém grafového neizomorfizmu a dokázali, že NONISO $\in$ **IP**.

Ďalej sme sa zaoberali vzťahom medzi týmito dvoma triedami. Dokázali sme slabšie tvrdenie, že triedy $\mathbf{IP}(poly)$ a $\mathbf{AM}(poly)$ sú ekvivalentné. Na záver sme dokázali, že pre každý polynóm Q, $\mathbf{IP}(Q) \subseteq \mathbf{AM}(Q+2)$.

Literatúra

[1] Babai L.: *Trading group theory for randomness*, Proc. of 17th Symposium on the Theory of Computation, Providence, Rhode Island, 1985.

[2] Babai L., Moran S.: *Arthur-Merlin Games: A Randomized Proof System, and a Hierarchy of Complexity Classes*, Journal of Computer and System Sciences, Vol. 36, issue 2, April 1988. 254–276.

[3] Goldreich O., Mansour Y., Sipser M., *Interactive proof systems: Provers that never fail and random selection*, 28th Annual Symposium on Foundations of Computer Science, Los Angeles, 1987. 449–461.

[4] Goldwasser S., Micali S.,Rackoff C.: *The Knowledge complexity of interactive proofs*, Proc. of 17th Symposium on the Theory of Computation, Providence, Rhode Island, 1985.

[5] Goldwasser S., Micali S., Wigderson A.: *Proofs that yield nothing but their validity*, Journal of the ACM, Vol. 38, issue 3, July 1991. 690–728.

[6] Goldwasser S., Sipser M.: *Private coins versus public coins in interactive proof systems*, Proc. of 18th Annual ACM Symposium on Theory of Computing, Berkeley, California, 1986. 59–68.

[7] Papadimitriou Ch.: *Computational Complexity*, Addison Wesley, 1994.

[8] *Advanced complexity theory*, http://people.csail.mit.edu/madhu/ST02/