Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Denis Vald

Důkaz s nulovou znalostí pro isomorfismus grafů

Katedra Algebry

Vedoucí bakalářské práce: Mgr. Štěpán Holub, Ph.D.

Studijní program: Matematika
Studijní obor: Matematické metody informační bezpečnosti

2008

Rád bych poděkoval svému vedoucímu, Mgr. Štěpánu Holubovi, Ph.D., za poskytnutou pomoc při vytváření této práce. Zároveň děkuji své rodině za duševní i fyzickou podporu a projevenou důvěru. Také bych chtěl poděkovat svým kolegům, se kterými jsem se při tvorbě práce radil. Tímto všem výše uvedeným srdečně děkuji.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 6. srpna 2008 Vald Denis

Obsah

1 Definice a základní vlastnosti **5**

1.1 Obecná představa 5
1.2 Technická představa 5
1.3 Turingův stroj 5
1.4 Interaktivní protokol 7
1.5 Zero-knowledge 9
1.6 Zesílení definice ZK 13

2 Příklady protokolů **17**

2.1 Příklady jazyků třídy IP. 17
2.2 Příklady jazyků třídy ZK. 19
2.3 ZK pro NP 20

3 Skládání protokolů **24**

3.1 Typy skládání a nedosažitelné množiny. 24
3.2 Sekvenční skládání. 26
3.3 Paralelní skládání 32

Literatura **36**

Název práce: *Důkaz s nulovou znalostí pro isomorfismus grafů*
Autor: *Denis Vald*
Katedra: *Katedra Algebry*
Vedoucí bakalářské práce: *Mgr. Štěpán Holub, Ph.D.*
e-mail vedoucího: holub@karlin.mff.cuni.cz

Abstrakt: V předložené práci zkoumáme tzv. důkazy s nulovou znalostí. S využitím teorie Turingových strojů definujeme interaktivní důkazové systémy, jejichž speciálním případem jsou právě důkazy s nulovou znalostí. Popíšeme několik různých definic daného pojmu a ukážeme si vztahy mezi nimi. Odlišnosti v různě silných podmínkách kladených v jednotlivých definicích nám umožňují dokázat důležité vlastnosti (mj. pro použití v praxi). Dále následují konkrétní příklady protokolů splňujících vlastnost nulové znalosti. Nakonec se zaměříme na sekvenční a paralelní skládání (tj. sériové a souběžné opakování jednotlivých protokolů). Cílem práce je seznámit čtenáře se známými výsledky daného tématu z širšího úhlu pohledu a tímto způsobem (aspoň částečně) odpovědět na původní otázku, za je možné skládat důkazy s nulovou znalostí.

Klíčová slova: důkaz s nulovou znalostí, sekvenční a paralelní skládání protokolů, isomorfismus grafů, černá skříňka

Title: *Zero-knowledge proofs for graph isomorphism*
Author: *Denis Vald*
Department: *The Department of Algebra*
Supervisor: *Mgr. Štěpán Holub, Ph.D*
Supervisor's e-mail address: holub@karlin.mff.cuni.cz

Abstract: In this work we examine so-called zero-knowledge proofs. Using the theory of Turing machines we first define interactive proof systems, which zero-knowledge proofs are special case of. We present several different definitions of the current notion and investigate their mutual relations. More restrictive requirements appear to be crucial in proving some important properties (that are used when applying ZK proofs in practice). Next, some specific examples of protocols satisfying the zero-knowledge property follow. In last chapter we focus on the sequential and parallel composition (i.e., repeating one execution after the other and running several executions simultaneously). Our aim is to present known results in the wider perspective, thus (at least partially) answering the original question whether composition preserves zero knowledge property.

Keywords: zero-knowledge proof, sequential and parallel composition of protocols, graph isomorphism, black-box

Kapitola 1

Definice a základní vlastnosti

1.1 Obecná představa

Pojem důkaz s nulovou znalostí (dále jen ZK od angl. *zero-knowledge proof*) by se mohl na první pohled zdát jako jistý oxymóron. Nejde však o snahu dokázat určitý fakt bez jakýchkoli (příslušných) znalostí; cílem je přesvědčit druhou stranu, že ‚něco' víme, aniž bychom však prozradili ono ‚něco' (znalost kterého by poté mohla být zneužita). Jak je patrné z předchozí věty, neobejdeme se bez přesných definic – zadefinujeme si, co je to komunikace, mezi kým probíhá, co znamená něco vědět a něco prozradit, apod. K tomu budeme potřebovat pojmy jako: (Polynomiální pravděpodobnostní) Turingův stroj a jeho interaktivní varianta, interaktivní protokol, důkaz s nulovou znalostí, atd.

1.2 Technická představa

Z technického hlediska ZK je interaktivní protokol probíhající mezi dvěma (interaktivními) Turingovými stroji. Ty si mezi sebou vyměňují zprávy, na konci komunikace jeden (přesně určený) stroj oznámí, jestli důkaz přijal či nikoliv. Stroje se označují jako P(eggy) a V(ictor), což je pouze personifikace pojmů Prover (Dokazovatel/Svědek) a Verifier (Ověřovatel/Sudí). Jak už názvy napovídají, stroj *P* se snaží dokázat stroji *V*, že má nějakou znalost, jenž je obvykle spojena s identitou nebo požadovaným chováním *P*, tedy můžeme pro zjednodušení říci, že jde o prokázání identity nebo poctivosti jednání. Přirozeně vznikají požadavky, aby ani *P* ani *V* nemohli podvádět (to je zachyceno v definici interaktivního protokolu), k tomu se poté přidává požadavek na nevyzrazení žádné informace (def. nulové znalosti).

1.3 Turingův stroj

Turingův stroj (TM od angl. *Turing machine*) je abstrakce vycházející z otázky, co je to algoritmus. V podstatě jde o matematický model klasického počítače. Dle Church-Turingovy teze dokonce každý představitelný algoritmus lze provádět na nějakém Turingově stroji (tj. pro každý algoritmus existuje Turingův stroj, který jej simuluje…).

Definice 0.0: Turingův stroj (Turing machine - TM) sestává z oboustranně nekonečné pásky rozdělené na políčka obsahující po jednom symbolu a čtecí hlavy, která ukazuje vždy na určité políčko pásky. Hlava může číst a zapisovat (resp. přepisovat) symboly a posouvat se po pásce doleva a doprava. Kromě toho má hlava svou vnitřní paměť, kde je uložen její stav.
Pro úplnost následuje podrobnější matematický popis:
Nechť je Σ množina symbolů, které se mohou vyskytovat na pásce. **Slovem** rozumíme uspořádanou n-tici symbolů ze Σ. Rovnou poznamenejme, že stačí $\Sigma = \{0;1\}$ (zapisovat všechno v binární abecedě). Dále mějme speciální symbol $\notin \Sigma$, který bude označovat

prázdné políčko. Σ* označuje množinu slov libovolné (konečné) délky: $\bigcup_{n\in\mathrm{N}}(a_1,...,a_n),\ a_i \in \Sigma, 1 \le i \le n$.

Vnitřní abeceda stroje: $\Gamma \supseteq \Sigma^* \cup \{\ \ \} \cup \ldots$ (může obsahovat další pomocné symboly; uvádíme pouze pro pořádek, nebudeme dále využívat). **Okamžitým stavem** stroje rozumíme obsah vnitřní paměti hlavy a symbol, na který hlava ukazuje. Množinu stavů označíme Q. Pro každý takový stav by měla existovat instrukce říkající stroji, co dál: zápis na pásku + změna vnitřní paměti hlavy + posunutí se. Stroj, resp. jeho program (což je množina instrukcí pro daný stroj), lze chápat jako přechodovou funkci:

$$T : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\leftarrow, \rightarrow, -\}$$
$$q \times s \ \mapsto\ q' \times s' \times \leftrightarrow$$

Výpočet stroje $\mathrm{M} = (\Sigma, Q, \Gamma, \mathrm{T})$.

Krátce řečeno: Hlava se dostane na určité políčko a podle jeho obsahu a svého vnitřního stavu vyvolá instrukci – tj. provede přepsání/změnu stavu/posune se. Pokud instrukce pro daný stav neexistuje, výpočet končí a na pásce je výstup.

Nedeterministický TM je takový stroj, kdy pro nějaký stav existuje více možných přechodů do dalšího stavu - přechodová funkce je relace; (zatímco u deterministického TM je pro každý stav nejvýše jedna instrukce). Pokud jsou nedeterministické přechody vyjádřeny pravděpodobností, pak mluvíme o **pravděpodobnostním TM**.

Vstupem a **výstupem** TM je nějaká (nepřerušovaná) posloupnost symbolů – nějaké slovo. **Jazykem** rozumíme libovolnou podmnožinu $L \subseteq \Sigma^*$.

Přirozeně nás zajímá otázka efektivnosti algoritmu/stroje. My se hlavně setkáme s **polynomiálními TM**, což jsou stroje, jejichž počet operací lze vyjádřit jako *pnom*(*n*), kde *n* je délka vstupu a *pnom* je nějaký polynom (v proměnné *n*). Pro pravděpodobnostní TM *pnom*(*n*) omezuje počet kroků nezávisle na pravděpodobnostních přechodech (tzn. omezení na počet kroků platí pro všechny možnosti). Takové výpočty obvykle považujeme za prakticky rychlé – pokud řekneme, že je stroj/algoritmus/výpočet **efektivní**, myslíme tím, že jde o pravděpodobnostní polynomiální stroj (nebo výpočet proveditelný takovým strojem).

Rozhodovací problém je úloha, v níž máme rozhodnout, jestli dané slovo patří do určitého jazyka. (Např. zjištění prvočíselnosti daného čísla lze formulovat jako rozhodovací problém, kde vstupem je zadané číslo a jazykem je množina prvočísel.) Pak pro TM, který je schopen řešit příslušný rozhodovací problém – rozhoduje daný problém, pišme: $x \in L \Rightarrow TM(x) = 1;\ \ x \notin L \Rightarrow TM(x) = 0$, a říkáme, že stroj přijal, resp. nepřijal vstup.

Rozšíříme-li základní definici TM o **více pásek** (např. budeme chtít mít zvlášť pásku vstupní, pracovní a výstupní), pak stroj vypadá tak, že každé pásce přísluší jedna čtecí hlava a ty fungují jako v základní definici. Množina stavů je pro všechny hlavy společná (ale může sestávat ze sjednocení stavů pro jednotlivé hlavy). Vícepáskový stroj však lze simulovat strojem jednopáskovým za cenu polynomiálního nárůstu počtu kroků.

Protože budeme potřebovat popsat vzájemnou komunikaci více (dvou) stojů, zadefinujeme si takovou komunikace-schopnou dvojici.

Definice 0.1: ***Interaktivní Turingův stroj*** *(ITM) je vícepáskový pravděpodobnostní TM, jehož pásky jsou: veřejná vstupní páska, soukromá vstupní páska, pracovní páska, vstupní a výstupní komunikační pásky, (veřejná) výstupní páska a stavový bit, což je jednopolíčková páska obsahující vždy buď 0, nebo 1. Vstupní pásky jsou pouze ke čtení,*

výstupní pouze k zápisu. Dále je každému stroji přiřazena tzv. identita $\sigma \in \{0; 1\}$. Stroj pracuje, právě když stavový bit se rovná jeho identitě, jinak je nečinný (obsah výstupních pásek se neměnní a čtecí hlavy se neposouvají).

Definice 0.1 si zaslouží několik poznámek:

V některých případech (speciálně např. v [3]) se ITM definuje jako deterministický a přidává se ještě jedna páska obsahující (nekonečnou) posloupnost náhodně zvolených bitů (tzv. náhodný vstup), podle které se také stroj při výpočtu řídí; pak ale lze interpretovat stroj jako pravděpodobnostní jako v Definici 0.1.

V ‚nejzákladnější' definici ITM nemají žádný soukromý vstup, pouze veřejný (společný). Jak však uvidíme dále při zkoumání vlastností nulové znalosti, toto by bylo pro naše potřeby nedostačující.

V této chvíli možná není Definice 0.1 úplně průhledná (např. jak nečinný stroj přejde do aktivního stavu) – vše se objasní tím, že my budeme vždy uvažovat dvojici interaktivních strojů.

Definice 1.1: ***Dvojice*** *ITM A, B se nazývá* ***propojená****, ozn. $\langle A, B\rangle$, pokud A a B mají opačnou identitu, sdílejí veřejnou vstupní pásku, sdílejí stavový bit, vstupní komunikační páska stoje A je výstupní komunikační páska stroje B a naopak. (Ostatní pásky jsou oddělené.)*

Tedy výpočet funguje tak, že přepisováním stavového bitu si stroje ‚udílejí slovo' a pomocí komunikačních pásek si ‚posílají zprávy'. Výstupem propojené dvojice $\langle A, B\rangle$ na společném vstupu x, ozn. $\langle A, B\rangle(x)$, rozumíme výstup stroje B po komunikaci se strojem A ($\langle A, B\rangle(x) = 1/0$, vstup x byl přijat/odmítnut). (Časová) náročnost $\langle A, B\rangle$ se posuzuje podle náročnosti stroje B, tj. stroj B se zastaví po počtu kroků rovnému náročnosti $\langle A, B\rangle$. Poznamenejme, že takto daná náročnost nezávisí na zprávách, které B přijal od A (tedy jde o horní odhad). Výše uvedené asymetrie (důraz na výstup a nároky stroje B) plyne z toho, že A chápeme jako dokazovatele, že $x \in L$, a B ověřovatele tohoto tvrzení.

1.4 Interaktivní protokol

Definice 1.2: ***Interaktivní důkazový systém*** *pro jazyk L je propojená dvojice $\langle P, V\rangle$ splňující následující podmínky:*

1) *V pracuje v polynomiálním čase (efektivita)*
2) $\forall x \in L : \Pr\left[\langle P, V\rangle(x) = 1\right] \geq \frac{2}{3}$ *(úplnost)*
3) $\left(\forall x \notin L\right)\left(\forall \text{ ITM } P^*\right) : \Pr\left[\langle P^*, V\rangle(x) = 1\right] \leq \frac{1}{3}$ *(korektnost)*

V Definici 1.2 efektivita znamená, že ověřovatel je z praktického hlediska efektivní. Úplnost nám zaručuje, že ‚správný' vstup bude ‚většinou' přijat, zatímco podmínka korektnosti zajišťuje, že ‚špatný' vstup bude ‚většinou' odmítnut. Dále je třeba dát pozor na to, že podmínka korektnosti se vztahuje na všechny možné stroje P^* (nejen na námi definovaného pro daný interaktivní důkazový systém). Někdy budeme P a V říkat strategie (místo algoritmus/stroj).

Hodnoty $\frac{2}{3}$ a $\frac{1}{3}$ nazýváme (dolní) mez úplnosti a (horní) mez korektnosti. Neostré nerovnosti nemají žádný hlubší význam, dokonce ani konkrétní hodnota mezí úplnosti a korektnosti není důležitá ve smyslu následující poznámky:

Mějme dvě funkce $u, k: \mathbb{N} \to \langle 0; 1 \rangle$.

Poznámka 1.3:

1) *(Tzv. všeobecný interaktivní důkazový systém) Nechť pro daný polynom $p(\cdot)$ a daný jazyk L existuje interaktivní důkazový systém takový, že funkce $u(\cdot)$ je mez úplnosti a funkce $k(\cdot)$ je mez korektnosti a platí $u(|x|) > k(|x|) + \frac{1}{p(|x|)}$. Dále nechť jsou tyto funkce počitatelné v polynomiálním čase. Potom pro tento jazyk L existuje interaktivní důkazový systém s mezí úplnosti $1 - 2^{-p(|x|)}$ a mezí korektnosti $2^{-p(|x|)}$.*
2) *Každý interaktivní důkazový systém lze transformovat tak, že mez úplnosti se bude rovnat 1.*

Důkaz první části by sestával z dostatečného počtu opakování daného inter. důk. systému, druhá část by vyžadovala hlubší studium.

Předchozí tvrzení osvětluje zmiňovanou potřebnou míru vstup ‚většinou' přijmout / odmítnout.

Definice 1.4*: IP* *(interactive proof) je třída jazyků L taková že, pro L existuje interaktivní důkazový systém.*

Pro lepší představu ukážeme základní souvislosti s jinými složitostními třídami.

Definice 1.6: *Třída **P** obsahuje jazyk L, pokud existuje polynomiální TM rozhodující tento jazyk.*

Definice 1.7: *Třída **NP**: jazyk $L \subseteq \Sigma^*$ je v NP, pokud:*

$(\exists \text{relace } R \text{ na } \Sigma^* \times \Sigma^*)(\exists c \geq 1)(\forall x \in \Sigma^*)$: i) $x \in L \Leftrightarrow \exists y \in \Sigma^* : |y| \leq |x|^c \wedge R(x, y)$

ii) $R(x, y) \in \boldsymbol{P}$

Tedy třída *NP* obsahuje takové jazyky, pro něž existuje relace taková, že pro každé slovo existuje svědek dokládající náležení daného slova do jazyka; ověření relace je možné provést polynomiálním TM (proto i podmínka na délku svědka). *NP* lze chápat jako třídu jazyků rozhodovanou polynomiálním TM, pokud dostane jako pomocný vstup příslušného svědka, proto zřejmě $P \subseteq NP$ (daný stroj nepotřebuje svědka).

Definice 1.5: *Třída **BPP** (bounded probabilistic polynomial time) osahuje takové jazyky L, že existuje pravděpodobnostní polynomiální stroj M splňující*:

1) $\forall x \in L : \Pr[M(x) = 1] \geq \frac{2}{3}$
2) $\forall x \notin L : \Pr[M(x) = 1] \leq \frac{1}{3}$

Jinými slovy *BPP* lze rozhodovat polynomiálním pravděpodobnostním strojem, který nechybuje příliš často (podobně jako u třídy *IP* lze snížit pravděpodobnost chyby, pokud máme TM chybující nejhůře v $\frac{1}{2} - \frac{1}{p(|x|)}$ případech)

Tvrzení 1.6:

1) BPP ⊆ IP
2) NP ⊆ IP

Důkaz:

1) Mějme $L \in BPP$ a stroj *M* rozhodující daný jazyk. Pak interaktivní důkazový systém pro *L* bude vypadat následovně: ⟨*O*, *M*⟩, kde *O* je *ITM*, který nic nedělá (alternativně – pošle prázdnou zprávu), a stroje *M* rozhodne náležení vstupu do jazyka. Tím je splněna definici *IP*.
2) Uvažujme následující dvojici ⟨*P*, *V*⟩ na vstupu *x: P* pošle $y \in \Sigma^*$; pokud je splněna svědecká relace *R*(*x*, *y*), pak *V* přijme, jinak odmítne; je-li $x \in L$, *P* díky své neomezené výpočetní síle svědka najde, jeho délka je nejvýše polynomiální (vzhledem k délce *x*) – takže ho může *V* zpracovat a svědecká relace *R* je počitatelná v polynomiálním čase, takže i toto *V* zvládne;
pro $x \notin L$ žádný svědek neexistuje, proto *R*(*x, y*) nebude splněna, ať je *y* jakékoliv □

V Tvrzení 1.6 jsme rozebrali dvě svým způsobem krajní situace – jednou ověřovatel nepotřebuje dokazovatele (*BPP*), v druhém případě dokazovatel poskytne veškeré potřebné informace pro ověření tvrzení (*NP*). Složitější příklady uvedeme v Kapitole 2. Pro představu velikosti třídy *IP* uvedeme (bez důkazu) ještě jeden fakt.

Definice 1.7: *Třída PSPACE obsahuje takové jazyky, že pro každý $L \subseteq PSPACE$ existuje TM rozhodující daný jazyk a polynom p(n) (n = |x|) tak, že stroj navštíví nejvýše p(n) políček. (Jinými slovy: PSPACE je analogie P vzhledem k prostorové složitosti.)*

Věta 1.8 (Shamir): *IP = PSPACE*

1.5 Zero-knowledge

Jednoduše řečeno – důkaz s nulovou znalostí je komunikace, kde ověřovatel se přesvědčí o správnosti tvrzení, ale dokazovatel neprozradí nic navíc. ‚Nic neprozradit' v tomto případě znamená, že cokoliv si ověřovatel může spočítat po interakci s *P*, si mohl spočítat sám (aniž by komunikoval s *P*).
Před zformalizováním právě vyslovených požadavků do definice nulové znalosti, potřebujeme zavést blízkost souborů náhodných veličin.

Definice 1.9: *Nechť jsou* $\{R_x\}_{x\in L}$ *a* $\{S_x\}_{x\in L}$ *soubory náhodných veličin; pak je nazveme*

a) ***identické****, pokud* $(\forall x \in L)(\forall y \in \Sigma^*) : \Pr[R_x = y] = \Pr[S_x = y]$

b) ***statisticky blízké****, pokud* $(\forall \text{polynom } p(n))(\exists n_0 > 0)(\forall n > n_0)(\forall x \in L,\ |x| \geq n)$*:*

$$\sum_{y\in\Sigma^*} \left|\Pr[R_x = y] - \Pr[S_x = y]\right| < \frac{1}{p(n)}$$

c) ***výpočetně nerozlišitelné**, pokud pro každý polynomiální pravděpodobnostní stroj D a platí:*

$$(\forall \text{polynom } p(n))(\exists n_0 > 0)(\forall n > n_0)(\forall x \in L,\ |x| \geq n):$$

$$\left|\Pr\left[D(x, R_x) = 1\right] - \Pr\left[D(x, S_x) = 1\right]\right| < \frac{1}{p(n)}$$

Technická poznámka: Budeme psát „pro dostatečně velká x je funkce $f(x)$ zanedbatelná funkce (v x)“ místo „$(\forall \text{polynom } p(n))(\exists n_0 > 0)(\forall n > n_0)(\forall x \in L,\ |x| \geq n): f(x) < \frac{1}{p(n)}$“

Definice 1.10*: **Důkazem s nulovou znalostí** nazveme interaktivní důkazový systém ⟨P, V⟩ pro jazyk L, pokud pro každý polynomiální ITM V* existuje (neinteraktivní) polynomiální pravděpodobnostní stroj M* takový, že platí dvě následující podmínky:*

1) $\forall x \in L: \Pr[M^*(x) = \perp] \leq \frac{1}{2}$, *kde symbol ⊥ označuje, že stroj M při výpočtu selhal (není schopen vydat 0 ani 1)*

2) *Následující soubory náhodných veličin*
   - $\{\langle P, V^* \rangle(x)\}_{x \in L}$
   - $\{M^*(x) \mid M^*(x) \neq \perp\}_{x \in L}$

   *a) jsou identické; pak mluvíme o důkazu s **dokonale** nulovou znalostí (třída jazyků **PZK**)*
   *b) jsou statisticky blízké; důkaz s **téměř dokonale** nulovou znalostí (**SZK**)*
   *c) jsou výpočetně nerozlišitelné; důkaz s **výpočetně** nulovou znalostí (**CZK**)*
   Stroj M se nazývá simulátor.

Tedy důkaz s nulovou znalostí je interaktivní důkazový systém, kde na dokazovatele je kladen doplňkový požadavek. A to takový, že libovolný ověřovatel si veškerou interakci s ním může nasimulovat sám, přičemž výsledek simulace se musí skutečné komunikaci podobat buď úplně (*PZK – perfect ZK*), nebo musí být statisticky blízko (*SZK - statistical*), anebo od ní nesmí být odlišitelný jakýmkoliv efektivním algoritmem (*CZK – computational*).

Protože z praktického hlediska nám většinou stačí výpočetní nerozlišitelnost („pokud nedokážeme dva objekty efektivně rozeznat, zdají se nám být stejné“), třída *CZK* se někdy označuje pouze jako *ZK*; nebude-li v textu výslovně uvedeno jinak, pojmem důkaz s nulovou znalost (*ZK*) budeme mít na mysli důkaz s výpočetně nulovou znalostí (*CZK*).

Všimněme si několika věcí: Výstup po komunikaci a výstup simulátoru jsou náhodné veličiny dané náhodností algoritmů *P*, *V**, *M**. Simulátor musí existovat pro jakéhokoliv ověřovatele, tj. ne nutně jenom pro řídícího se předepsaným protokolem, právě naopak – hlavní nebezpečí představují *V** snažící se od dokazovatele získat nějakou znalost svým ‚záludným‘ chováním. Dále se vyžaduje, aby *M** (neselhával příliš často a) generoval podobný soubor, ale pouze pro vstupy $x \in L$, tj. tato jediná informace má být prozrazena. Poznamenejme také, že ač to není v Definici 1.10 výslovně uvedeno, platí definice i pro všeobecný interaktivní důkazový systém (tj. takový, který má nezanedbatelný rozdíl mezi mezemi úplnosti a korektnosti).

Obdobně jako u Poznámky 1.3 bychom mohli povolit/požadovat selhání M^* s pravděpodobností $(1-\frac{1}{p(n)})$ / $(2^{-p(n)})$; zesílení bychom dosáhli polynomiálním opakováním simulace. Dokonce podmínku 1) v Definici 1.10 pro třídy *SZK* a *CZK* můžeme zanedbat, aniž bychom změnili velikost třídy, přesněji:

Poznámka 1.11: *Nechť existuje simulátor M* plně vyhovující Definici 1.10 pro třídy SZK a CZK. Potom existuje simulátor M**, jemuž není dovoleno selhat (podmínka 1 Definice 1.10) a přitom stále produkuje statisticky blízký/výpočetně nerozlišitelný soubor náhodných veličin.*

Idea důkazu: M^{**} bychom sestrojili tak, že by polynomiálně-krát zavolal M^* a vydal by jeho výstup, pokud by M^* byl úspěšný, jinak by také selhal (můžeme si představit, že v tomto případě by jeho výstup byl zvolen náhodně s pravděpodobností jedna polovina). Díky tomu by pravděpodobnost selhání M^{**} byla exponenciálně malá (v $|x|$) a tyto případy by nenarušili statistickou blízkost/výpočetní nerozlišitelnost; v případě *PZK* by však případy selhání nezachovaly identičnost souborů. Podrobnosti viz [9].

Výše jsme uvažovali rozdíl mezi výstupem ověřovatele po komunikaci s dokazovatelem a výstup (simulátoru) bez žádné komunikace. Přirozenější by se mohlo zdát uvažovat veškeré informace, které ověřovatel obdrží, a na jejichž základě by se mohl dopracovat jemu nedostupné znalosti. Z tohoto hlediska by se definice nulové znalosti změnila takto:

Definice 1.12: *Nechť jsou $\langle P,V\rangle$, L, V* jako v Definici 1.10. Označíme $view^P_{V^*}(x)$ náhodnou proměnnou popisující obsah pásky náhodného vstupu V* a zprávy, které V* obdrží od P během komunikace na vstupu x. $\langle P, V\rangle$ nazveme* ***důkazem s nulovou znalostí****, pokud pro každý polynomiální ITM V* existuje (neinteraktivní) polynomiální pravděpodobnostní stroj M* takový, že soubory náhodných veličin $view^P_{V^*}(x)$ a $\{M^*(x)\}_{x\in L}$ jsou výpočetně nerozlišitelné. Analogicky pro (téměř) dokonalou nulovou znalost bychom požadovali shodnost (statistickou blízkost) výše uvedených souborů.*

Popis pásky náhodného vstupu je zapotřebí, pokud uvažujeme deterministické ITM (viz komentáře za Definicí 0.1). Definice 1.10 a Definice 1.12 se liší jen záměnnou $\langle P,V^*\rangle(x)$ za $view^P_{V^*}(x)$, z čehož plyne, že simulátory pro jednotlivé definice nejsou shodné, ačkoli jsou určitým způsobem spjaté. Zřejmě ze záznamu komunikace zjistíme (v polynomiálním čase) výstup dvojice $\langle P,V^*\rangle$ (tedy Definice 1.12 $\Rightarrow$ Definice 1.10). Platnost obrácené implikace bychom ukázali tak, že díky kvantifikaci přes všechny V^* existuje V^{**} splňující $view^P_{V^*}(x) = \langle P, V^{**}\rangle(x)$. Tedy tyto dvě definice jsou ekvivalentní a při dokazování ZK vlastnosti můžeme využívat jakoukoli z nich.

Pozorování 1.13: $BPP \subseteq PZK \subseteq SZK \subseteq CZK \subseteq IP$.

Důkaz:

$BPP \subseteq PZK$ – uvažujme interaktivní důkazový systém $\langle O, M\rangle$ jako v důkazu Tvrzení 1.6; pak *O* jako dokazovatel nic neprozradí, protože neposílá žádnou zprávu.

PZK $\subseteq$ *SZK* – *PZK* důkaz splňuje podmínky *SZK*, protože identické soubory mají statistickou odchylku rovnou nule

SZK $\subseteq$ *CZK* – podobně jako výše stačí ukázat,že statistická blízkost implikuje výpočetní nerozlišitelnost – viz Lemma 1.14

CZK $\subseteq$ *IP* – splněno triviálně (dle Definice 1.10 $\langle P, V\rangle \in IP$) □

Lemma 1.14: *Nechť jsou* $\{R_x\}_{x\in L}$ *a* $\{S_x\}_{x\in L}$ *statisticky blízké soubory náhodných veličin; pak jsou tyto soubory výpočetně nerozlišitelné.*

Důkaz: (dle návodu [3])

1) Nejprve dokážeme, že statistická blízkost implikuje následující: pro každé dostatečně dlouhé $x\in L$ a pro každou množinu $N \subseteq \Sigma^*$ platí

$$\left|\Pr[R_x \in N] - \Pr[S_x \in N]\right| < \frac{1}{p(|x|)};$$

Výše uvedené platí, protože:

$$\left|\Pr[R_x \in N] - \Pr[S_x \in N]\right| = \left|\sum_{y\in N}\Pr[R_x = y] - \sum_{y\in N}\Pr[S_x = y]\right|$$

$$\left|\sum_{y\in N}\Pr[R_x = y] - \sum_{y\in N}\Pr[S_x = y]\right| = \left|\sum_{y\in N}\left(\Pr[R_x = y] - \Pr[S_x = y]\right)\right|;$$

teď bychom chtěli shora odhadnout poslední výraz: $\Pr[\ldots] \geq 0$. Suma v absolutní hodnotě bude nabývat svého maxima pokud všechny sčítance budou stejného znaménka. Označme tedy $\forall x \in L$

$$M_x^{pos} = \left\{y \in \Sigma^*; \left(\Pr[R_x = y] - \Pr[S_x = y]\right) > 0\right\}$$

$$M_x^{neg} = \left\{y \in \Sigma^*; \left(\Pr[R_x = y] - \Pr[S_x = y]\right) < 0\right\}$$

$$M_{\max} = \max\{M_x^{pos}, M_x^{neg}\};$$

přechod od $y \in N$ k $y \in \Sigma^*$ pouze zvýší horní odhad; tedy

$$\left|\sum_{y\in N}\left(\Pr[R_x = y] - \Pr[S_x = y]\right)\right| \leq \left|\sum_{y\in M_{\max}}\left(\Pr[R_x = y] - \Pr[S_x = y]\right)\right|$$

$$\left|\sum_{y\in M_{\max}}\left(\Pr[R_x = y] - \Pr[S_x = y]\right)\right| = \sum_{y\in M_{\max}}\left|\Pr[R_x = y] - \Pr[S_x = y]\right|;$$

poslední rovnost máme právě díky definici množiny $M_{\max}$ (stejná znaménka sčítanců); nakonec z $M_{\max} \subseteq \Sigma^*$ a předpokladu platí

$$\sum_{y\in M_{\max}}\left|\Pr[R_x = y] - \Pr[S_x = y]\right| \leq \sum_{y\in \Sigma^*}\left|\Pr[R_x = y] - \Pr[S_x = y]\right| < \frac{1}{p(|x|)},$$

což jsme chtěli ukázat.

2) Teď vhodnou volbou množiny *N*:
$\forall f : \Sigma^* \to \{0;1\}$ nechť $N_f = \{x \in \Sigma^*; f(x) = 1\}$; tedy z 1) plyne:
Nechť jsou $\{R_x\}_{x\in L}$ a $\{S_x\}_{x\in L}$ statisticky blízké soubory náhodných veličin, pak

$$\frac{1}{p(|x|)} > \left|\Pr[R_x \in N_f] - \Pr[S_x \in N_f]\right| = \left|\Pr[f(R_x) = 1] - \Pr[f(S_x) = 1]\right|$$

Teď už si jen stačí uvědomit, že každý polynomiální pravděpodobnostní TM *D*, který by měl rozlišovat dva soubory náhodných veličin, si lze představit jako funkci $f : \Sigma^* \to \{0;1\}$, tedy místo $\Pr[f(R_x) = 1]$ lze psát $\Pr[D(x, R_x) = 1]$ (*x* stroji *D* poskytujeme kvůli požadavku polynomiality). Tím máme dokázanou výpočetní nerozlišitelnost. □

1.6 Zesílení definice ZK

Doteď jsme se dívali na důkazy s nulovou znalostí jako na komunikaci, kde cokoliv lze (efektivně) spočítat po komunikaci s dokazovatelem, lze (efektivně) spočítat i pouze ze samotného *společného* vstupu. Tento pohled však nereflektuje plně praxi, kde ZK důkazy jsou využívány jako dílčí součásti větších protokolů. Ověřovatel jako potenciálně nepřátelský program by mohl získat nějakou pomocnou informaci z jiné části protokolu a i tu využít při komunikaci s dokazovatelem. Na druhou stranu obvykle ověřovatel v praxi nedisponuje neomezenou výpočetní silou a své ‚schopnosti' čerpá z nějaké pomocné znalosti, jež není dostupná ověřovateli. V tomto okamžiku se vracíme k zavedení pojmu interaktivní TM (Definice 0.1), kde jsme každému stroji povolili mít vlastní soukromý vstup. Ten právě reflektuje dodatečnou informaci, kterou může stroj při interakci mít – říkáme jí pomocný vstup a stroj *M* pracující s pomocným vstupem *w* označujeme *M*(*w*). Potřebujeme tedy zesílit Definice 1.2 a 1.10 v tom směru, že dokazovatel s pomocným vstupem nesmí být schopen ‚podvést' ověřovatele a ten naopak nesmí být schopen od dokazovatele zjistit žádnou znalost ani s pomocí svého soukromého vstupu.

Definice 1.15: ***Interaktivním důkazovým systémem vzhledem k pomocnému vstupu*** *pro jazyk L nazveme dvojici ⟨P, V⟩ splňující*:

1) V pracuje v čase polynomiálním vzhledem ke společnému vstupu x (efektivita)

2) $(\forall x \in L)(\exists y \in \Sigma^*)(\forall z \in \Sigma^*)\colon \Pr\left[\langle P(y), V(z)\rangle(x) = 1\right] \geq \frac{2}{3}$ *(úplnost)*

3) $(\forall x \notin L)(\forall \text{ITM } P^*)(\forall y, z \in \Sigma^*)\colon \Pr\left[\langle P^*(y), V(z)\rangle(x) = 1\right] \leq \frac{1}{3}$ *(korektnost)*

Řetězce *y*, *z* představují pomocný vstup pro *P*, *V*. Poznamenejme, že složitost (stroje *V*) se odvíjí od společného vstupu *x*, tedy polynomiální stroj se zastaví nejvýše po $p(|x|)$ krocích nezávisle na ostatním obsahu pásek, pro nějaký polynom $p(\cdot)$. V důsledku toho např. stroj nemusí být schopen přečíst celý svůj pomocný vstup.

Definice 1.16: *Nechť ⟨P, V⟩ je interaktivním důkazovým systémem vzhledem k pomocnému vstupu pro jazyk L (dle Definice 1.15). Označme* $P_L(x)$ *množinu y splňující podmínku úplnosti vzhledem k* $x \in L$ *(tzn.* $\Pr[\langle P(y), V(z)\rangle(x) = 1] \geq \frac{2}{3}, \forall z \in \Sigma^*$*).*

Důkazem s nulovou znalostí vzhledem k pomocnému vstupu *nazveme dvojici* $\langle P, V\rangle$*, pokud pro každý polynomiální ITM V* existuje (neinteraktivní) pravděpodobnostní stroj M* běžící v polynomiálním čase vzhledem ke svému prvnímu vstupu takový, že následující soubory náhodných veličin jsou výpočetně nerozlišitelné:*

- $\left\{\left\langle P(y_x), V^*(z)\right\rangle(x)\right\}_{x\in L; z\in\Sigma^*} \quad \forall y_x \in P_L(x)$
- $\left\{M^*(x,z)\right\}_{x\in L; z\in\Sigma^*}$

Tedy pro každý pravděpodobnostní stroj D běžící v polynomiálním čase vzhledem ke svému prvnímu vstupu, pro každý polynom $p(\cdot)$ *a pro všechna dostatečně dlouhá* $x\in L$*, všechna* $y_x \in P_L(x)$ *a* $z \in \Sigma^*$ *platí*

$$\left|\Pr\left[D\left(x,z,\left\langle P(y_x), V^*(z)\right\rangle(x)\right)=1\right]-\Pr\left[D\left(x,z,M^*(x,z)\right)=1\right]\right| < \frac{1}{p(|x|)}.$$

V Definici 1.16 představuje pomocný vstup pro dokazovatele řetězec *y*, pro ověřovatele řetězec *z* (který je také poskytnut rozlišovacímu stroji *D*). Zdůrazněme, že podle Definice 1.15 ověřovatel *V**, simulátor *M** i rozlišovatel *D* běží v čase polynomiálním vzhledem k (společnému vstupu) *x*. Nicméně polynom omezující běh rozlišovatele *D* (resp. přímo samotný stroj *D*) je volen až po zafixování polynomiálního omezení simulátoru *M**.

Definice 1.16 je definicí důkazu s výpočetně nulovou znalostí (vzhledem k pomocnému vstupu). Pro (téměř) dokonalou nulovou znalost by definice byla přímočaře upravena požadavkem na podobnost příslušných souborů náhodných veličin.

Poznámka 1.17: *V Definice 1.16 bychom místo rozlišovacího stroje D mohli uvažovat posloupnost polynomiálně velkých Booleovských obvodů (viz Definici 3.1 a Lemma 3.10.3). Na druhou stranu by nestačilo místo strojů V a M s pomocným vstupem použít posloupnosti obvodů, protože by nám to nezajistilo jednoduchou transformaci rozlišovatele na simulátor (zatímco u strojů toto implicitně vychází z konečnosti objektů) a také nemáme vztah mezi neuniformní částí V a odpovídající částí M (zatímco u strojů je toto provázání zajištěno neměnným pomocným vstupem).*

Další zesílení spočívá v zaměření se na simulátor. Požadovali jsme, aby pro každého ověřovatele takový existoval. Otázkou ovšem je, jak pro nekonečně mnoho strojů *V** ukázat konstrukci simulátoru *M**. Až na speciálně pro tyto účely konstruované příklady (viz [1]) je tento problém uchopen tak, že se pro všechny ověřovatele používá jeden univerzální simulátor, který má přístup ke každému ověřovateli jako k černé skříňce (orákulu). Představme si to tak, že univerzální simulátor má program stroje *V** a může ho pustit pro zvolený vstup a poté přečíst výstup (aniž by ‚viděl' dovnitř programu).

Definice 1.18: *Nechť pro ITM B s veřejným vstupem x, pomocným vstupem z a náhodným vstupem r, funkce* $B_{x,z,r}(\cdot)$ *popisuje zprávu odesílanou B tak, že* $B_{x,z,r}(m^*)$ *označuje zprávu odeslanou B na veřejném vstupu x, pomocném vstupu z, náhodném vstupu r a přijaté posloupnosti zpráv m*. Pro jednoduchost nechť výstup B se objeví jako jeho poslední odeslaná zpráva.*
Říkáme, že pravděpodobnostní polynomiální stroj M je simulátor typu černá skříňka pro dokazovatele P a jazyk L, pokud pro každý pravděpodobnostní polynomiální stroj B, pro každý pravděpodobnostní polynomiální stroj D, každý polynom $p(\cdot)$*, všechna dostatečně dlouhá* $x\in L$ *a všechna* $z, r \in \Sigma^*$*:*

$$\left|\Pr\left[D^{B_{x,z,r}}\left(\langle P, B_r(z)\rangle(x)\right)=1\right]-\Pr\left[D^{B_{x,z,r}}\left(M^{B_{x,z,r}}(x)\right)=1\right]\right|<\frac{1}{p(|x|)},$$

kde $B_r(z)$ *označuje interakci stroje B na pomocném vstupu z a náhodném vstupu r.*
Nakonec interaktivní důkazový systém $\langle P, V\rangle$ *je* ***důkaz s nulovou znalostí se simulátorem typu černá skříňka****, pokud pro něj existuje simulátor typu černá skříňka.*

Tedy předchozí definice říká, že máme jeden univerzální simulátor, který napodobuje interakci s dokazovatelem pro libovolného ověřovatele, přičemž dovolujeme simulátoru přistupovat k ověřovateli jako k orákulu (tj. pokládat mu dotazy a přijímat odpovědi). A odlišit simulaci od skutečné komunikace není schopen žádný (efektivní) stroj ani s pomocí volání výše uvedeného orákula.

Pro jednoduchost budeme psát: *AI* / *BB* ZK pro důkazy s nulovou znalostí vzhledem k pomocnému vstupu/se simulátorem typu černá skříňka (z angl. auxiliary-input / black-box). Zřejmě *BB* ZK implikuje *AI* ZK (univerzální simulátor nám poskytuje jednotlivé simulátory pro každého ověřovatele).

Poznámka 1.19: *V Definici 1.18 rovnou mluvíme o black-box simulátoru, který je univerzální pro ověřovatele vzhledem k pomocnému vstupu. Mohli bychom však (např. pro zjednodušení zápisu) předpokládat existenci univerzálního simulátoru* M_u *pro obyčejné ověřovatele. S pomocí* M_u *lze sestavit pro každého ověřovatele simulátor* M_u*, který by dostal pomocný vstup ověřovatele a vyprodukoval opět výpočetně neodlišitelný soubor náhodných veličin (viz Tvrzení 1.20). Toto mj. znamená, že každý ZK protokol, ve kterém simulátor pracuje tak, že používá program ověřovatele jako černou skříňku, lze transformovat na AI ZK protokol.*

Tvrzení 1.20: *Nechť* $\langle P, V\rangle$ *je interaktivní důkazový systém takový, že existuje (jeden univerzální) stroj* M_u *, který pro každé* $x \in L$ *a každý ITM V' je schopen simulovat interakci P, V' (tzn. vyprodukovat výpočetně neodlišitelný soubor). Pak pro každý ITM V* a pomocný vstup y lze zkonstruovat stroj* M_{V^*}*, který simuluje interakci P, V*(y).*

Důkaz (dle [7]): Označme *Time*(*A*,*B*,*x*) čas běhu (počet jednotkových operací) stroje *B* při komunikaci s *A* na vstupu *x* (vzhledem k tomu, že dle Definice 1.16 časová náročnost je nezávislá na pomocném vstupu, toto označení platí i pro stroje využívající pomocný vstup).
Popišme konstrukci simulátoru M_{V^*} pro libovolný stroj *V** s pomocným vstupem *y*: mějme polynom $Q(|x|) \geq Time(P, V^*, x)$ (existence plyne z polynomiality *V**). M_{V^*} bude vícepáskový stroj se zabudovaným kódem *V** (tj. schopný na vstupu *x, y* vydat výstup *V*(x,y)*) a přístupem k M_u (univerzální simulátor z předpokladů). Na vstupu (*x*, *y*) stroj M_{V^*} vytvoří stroj V_y^* zabudováním části *y*, $|y| \leq Q(|x|)$, do stroje *V** – tj. V_y^* zkopíruje *y* na vstupní pásku a spustí *V**(*x,y*). Pro zprávu „SEND AI“ V_y^* pošle *y* (tohoto později využijeme u odlišovacích strojů, pro simulaci nemá význam). Sestrojením V_y^* už stroj M_{V^*} může spustit simulaci – funguje jako M_u s (black-box) přístupem k V_y^* a vydá výstup M_u. Jenom k němu ještě musí přidat *y* (abychom měli kompletní záznam komunikace). Výše uvedený popis lehce shrneme: Stroji M_u vytvoříme přístup k V_y^*, který zachycuje chování *V** i s použitím pomocného vstupu *y*. Ověříme podmínky kladené na simulátor:

Pozorování 1.20.1: *M_{V^*} pracuje v polynomiálním čase vzhledem k prvním vstupu.*

Důkaz: M_{V^*} simuluje výpočet M_u přistupující k orákulu V_y^*. Dle předpokladů je počet kroků M_u polynomiální v $|x|$ bereme-li volání V_y^* jako jednotkovou operaci Označme tento polynom $Q_M(|x|)$. Výpočet $V_y^*(x)$ je v podstatě výpočet $V^*(x,y)$, který je omezen polynomem $Q(|x|)$. Taktéž pracujeme pouze s částí y, jenž je nejvýše rovna $Q(|x|)$. Tedy celkově simulace $V_y^*(x)$ je omezena nějakým polynomem $Q_V(|x|)$. Nakonec výsledný čas M_{V^*} je omezen hodnotou $Q_M(|x|)\cdot Q_V(|x|)$, tedy polynomem. ∟

Pozorování 1.20.2: *Soubory $\left\{\left\langle P.V^*(y)\right\rangle(x)\right\}_{x\in L;\,y\in\Sigma^*}$ a $\left\{M_{V^*}(x,y)\right\}_{x\in L;\,y\in\Sigma^*}$ jsou výpočetně nerozlišitelné.*

Důkaz: Sporem – nechť existuje polynom $p(\cdot)$, stroj D a nekonečná posloupnost dvojic (x, y) označená S tak, že platí:

$$\forall(x,y)\in S:\left|\Pr\left[D\left(x,y,\left\langle P,V^*(y)\right\rangle(x)\right)=1\right]-\Pr\left[D\left(x,y,M_{V^*}(x,y)\right)=1\right]\right|<\frac{1}{p\left(|x|\right)}$$

Ukážeme, že v tomto případě existuje polynom Q(·), stroj *D'* a nekonečná posloupnost S dvojic (x, V_y^*) tak, že:

$$S'\subseteq\left\{(x,V_y^*);\,x\in L, Time(P,V_y^*,x)\le Q\left(|x|\right)\right\}\quad\wedge$$

$$\forall(x,V_y^*)\in S':\left|\Pr\left[D'^{V_y^*}\left(\left\langle P,V_y^*\right\rangle(x)\right)=1\right]-\Pr\left[D'^{V_y^*}\left(M_u^{V_y^*}(x)\right)=1\right]\right|<\frac{1}{p\left(|x|\right)}$$

A dojdeme ke sporu s existencí univerzální ho simulátoru M_u fungujícího i pro stroj V_y^* (popsaného výše). Z konstrukce V_y^* je zřejmé jeho polynomiální omezení. Sestavení stroje *D'* je následující: nejprve pošle stroji V_y^* zprávu „SEND AI“ (připomeňme, že dle Definice 1.18 rozlišovatel má black-box přístup k ověřovateli), čímž získá y. Poté spustí pro rozlišení stroj $D(x, y, \Psi)$ (kde Ψ *je z* $\left\langle P,V_y^*\right\rangle(x)$ nebo $M_u^{V_y^*}(x)$) a vydá jeho výstup. Vzhledem k tomu, že V_y^* pracuje jako $V^*(x,y)$ (až na technické odlišnosti) a M_{V^*} pracuje jako M_u s (black-box) přístupem k V_y^* (až na počáteční vytvoření V_y^*), bude stroj *D'* schopen odlišovat pro dvojice (x, V_y^*), pro něž D odlišuje příslušné dvojice (x, y).

∟

□

Kapitola 2

Příklady protokolů

V předchozí kapitole jsme si nadefinovali některé pojmy a zkoumali jejich vlastnosti v obecné rovině. Neukázali jsme však žádný konkrétní příklad, dokonce jsme ani nedokazovali, jestli vytvořené objekty vůbec existují (až na některé jednoduché případy). Nejprve předvedeme jazyky z třídy *IP*, dále se podíváme, jestli patří i do (některé třídy) *ZK*. Nakonec se podíváme i na vztah tříd *ZK* a *NP*.

Definice 2.1: *Grafem nazveme dvojici vrcholů a hran:* $G=(V,E);\ \ V=\{1,\ldots,n\}$, $E\subseteq\{\{i,j\};i,j\in V\}$. *Tedy (pro jednoduchost) vrcholy grafu označujeme prvky množiny* $[n]=\{1,\ldots,n\}$, *hrany bereme neorientované a nenásobné.*

Definice 2.2: *Mějme dva grafy* $G_0=(V_0,E_0)$ *a* $G_1=(V_1,E_1)$*; nazveme je isomorfní, pokud existuje bijekce* $\pi:V_0\to V_1$ *taková, že platí* $\{u,v\}\in E_0 \Leftrightarrow \{\pi(u),\pi(v)\}\in E_1$*; pak píšeme* $G_0\simeq G_1$ *(isomorfní) a* $G_0=\pi(G_1)$ *(je isomorfní dle isomorfismu* π*).*

Definice 2.3: *Problém grafového isomorfismu:* $\forall G_0,G_1:(G_0,G_1)\in GI \quad\equiv\quad G_0\simeq G_1$*;*
obdobně problém grafového neisomorfismu $\forall G_0,G_1:(G_0,G_1)\in GNI \quad\equiv\quad G_0\neq G_1$

Poznámka 2.4: *Rozhodnout, zda-li jsou dva grafy isomorfní je obecně těžký problém. Tj. není znám žádný polynomiální algoritmus, který by tento problém řešil. Na druhou stranu, pokud nám někdo dá příslušný isomorfismus, je snadné tuto bijekci ověřit. Vidíme tedy, že* $GI\in NP$*. (pro grafy s různými omezujícími podmínkami, např. rovinné grafy nebo grafy s omezeným stupněm všech vrcholů, existují polynomiální algoritmy rozhodující GI'). O neisomorfismu však není známo ani to, zda* $GNI\in NP$*. To, že jsou to vůbec rozhodnutelné problémy (existuje nějaký algoritmus na jejich rozhodování) plyne triviálně z možnosti vyzkoušet nejhůře všechny bijekce, kterých je konečný počet (*$n!$*).*

2.1 Příklady jazyků třídy IP

Budeme chtít ukázat, že GNI ∈ IP a GI ∈ IP. K tomu musíme předvést dvojici ⟨P, V⟩ splňující podmínky Definice 1.2.

Technická poznámka: Tvrzení dokazující vlastnosti určité propojené dvojici ⟨*P*, *V*⟩ budou záměrně číslována stejně jako konstrukce popisující průběh komunikace dané dvojice.

Konstrukce 2.5: *Budeme uvažovat dvojici ⟨P, V⟩, která má na společném vstupu* $(G_0,G_1)\in GNI$*. Oba grafy jsou na n vrcholech (Jinak by byly zřejmě neisomorfní). Interakce probíhá následovně:*

Kroky 1) a 2) se opakují dvakrát

1) *V zvolí náhodně* $i \in \{0;1\}$ *;*
   zvolí náhodně $\pi \in S_n$ *(množina permutací n-prvkové množiny);*
   zkonstruuje $H = \pi(G_i)$ *;*
   odešle H dokazovateli P;
2) *P pošle* $j \in \{0;1\} : H \simeq G_j$ *;*
   - *díky neomezené výpočetní síle j vždy najde*
   - *pokud by platilo* $H \simeq G_0 \simeq G_1$, *P zvolí j náhodně*
   - *pokud by* $G_0 \not\simeq H \not\simeq G_1$ *pak P vydá* $\perp$ *(jakkoli označí chybu v komunikaci)*
3) *V důkaz P přijme, pokud* $j = i$ *v obou kolech.*

Tvrzení 2.5: $GNI \in IP$.

Důkaz: Zkontrolujeme podmínky Definice 1.2. *V* opravdu může být implementován v pravděpodobnostním polynomiálním čase (grafy popíšeme maticemi incidence – n^2 symbolů, isomorfismus – n symbolů; a *V* nebude muset udělat víc kroků, než je schopen).
Úplnost: Pokud $G_0 \not\simeq G_1$, pak *H* je jednoznačně určen (a *P* ho najde díky své síle). V obou kolech bude platit $j = i$ a *V* přijme s pravděpodobností 1.
Korektnost: $G_0 \simeq G_1 \Rightarrow H \simeq G_0 \simeq G_1$ tedy *P**, který by chtěl podvádět nebude vědět jaké $i \in \{0;1\}$ *V* zvolil. Ani to nemůže nijak spočítat, protože *H* nenese informaci, od kterého *G* byl vytvořen. H je totiž graf náhodně volený z množiny s opakováním $M_i = \{\pi(G_i); \pi \in S_n\}$ (množina všech grafů isomorfních G_i, dovolujeme prvkům se vícekrát opakovat). Z isomorfnosti G_0 a G_1 plyne, že $M_0 = M_1$, tedy *H* je náhodně zvolený prvek M_i a pro libovolný stroj *P** bude *j* nezávislé na *i*. Tedy vzhledem k náhodné volbě *i*, $\Pr[j = i] = \frac{1}{2}$. Toto platí v každém kole a výsledně:

$$\Pr[\langle P^*, V \rangle(x) = 1] = \frac{1}{4}.$$

□

Na Konstrukci 2.5 vidíme, že (polynomiálně mnoha) opakováními protokolu lze pravděpodobnost chyby exponenciálně přiblížit 0. Také si všimněme, že ověřovatel opravdu využívá své neomezené síly (pro ‚hledání' isomorfismu k daným grafům).

Tvrzení 2.6: $GI \in IP$.

Důkaz: Důsledek Tvrzení 1.6 a Poznámky 2.4 – jednoduše dokazovatel odešle ověřovateli svědka, tj. isomorfismus π. □

Jak už jsme popsali v Tvrzení 1.6, konstruování interaktivních důkazových systémů pro jazyky z třídy NP je přímočaré. Konstrukce 2.5 nám však dává příklad, který ukazuje, že IP je opravdu velikostně netriviální třída (i když toto je přímo popsáno ve Větě 1.8).

2.2 Příklady jazyků třídy ZK

Obdobně jako výše se budeme zabývat, jestli $GNI, GI \in (P)ZK$, tedy jestli existuje dvojice $\langle P,V\rangle$, kde nejenže dokazovatel přesvědčí ověřovatele, ale přitom mu neprozradí žádnou znalost navíc.

Problém 2.7: $GNI \overset{?}{\in} PZK$

Rozbor: Uvažujme propojenou dvojici jako v Konstrukci 2.5. Ta splňuje podmínky třídy *IP*. Jak je to s vlastností nulové znalosti? Pro jakéhokoli *poctivého* V^* ověřovatele je snadné sestrojit simulátor M^*, který bude schopen napodobit komunikaci tohoto ověřovatele s P. M^* bude pracovat jako V^* s tím rozdílem, že musí zajistit vlastní generování zpráv od P. To udělá snadno – P (dle Definice 1.10 simulace probíhá pro $x \in L$) vždy oznámí $j = i$ a tato hodnota je M^* dostupná z předcházející simulace. Jenže požadavek ZK musí P splňovat (hlavně) při komunikaci s libovolným (nepřátelským) ověřovatelem (snažícím se získat od P nějakou znalost). A tady Konstrukce 2.5 nestačí: vezměme V^*, který by měl nějaký graf H, o němž by věděl, že je isomorfní jednomu z dvojice G_0, G_1 (popř. by prostě vygeneroval nějaký náhodný graf). Pokud tento graf pošle dokazovateli, dozví se něco, co sám nasimulovat nedokáže (isomorfismus polynomiální pravděpodobnostní stroj hledat neumí). Je tedy třeba donutit nepoctivého ověřovatele dokázat, že zná isomorfismus π (z Konstrukce 2.5), na druhou stranu musí být skrytý v případě, kdy $G_0 \simeq G_1$ (to zase pro případ, že by podváděl dokazovatel – podmínka korektnosti). V [6] je to řešeno vytvářením pomocných dvojic závislých na π, kterými se ověřovatel ‚zaváže ke znalosti π' . Vidíme, že při dokazování vlastnosti nulové znalosti je potřeba opravdu uvažovat všechny možné strategie ověřovatele.
(Výsledkem [6] je, že $GNI \in PZK$, pouze však pro polynomiální čas v průměru, zatímco pro striktně polynomiální čas neumíme ani dokázat $GNI \in SZK$).

Jiný problém použití v praxi tkví v tom, že se opíráme o neomezenou výpočetní sílu dokazovatele. Velice užitečné v tomto směru se jeví právě problémy z NP, kde můžeme neomezenou sílu transformovat do znalosti svědka, kterého ale musíme na druhou stranu tajit a tedy nemůžeme přímo použít k důkazu.

Tvrzení 2.8: $GI \in PZK$.

Důkaz(dle[8]): Nemohli bychom použít dvojici $\langle P,V\rangle$ z Tvrzení 2.6, protože prozrazení svědka zřejmě poruší vlastnost nulové znalosti. Vyjdeme z následujícího postupu:

Konstrukce 2.8*: Budeme uvažovat dvojici $\langle P,V\rangle$, která má na společném vstupu (G_0, G_1). $G_0 = (V_0, E_0)$; $G_1 = (V_1, E_1)$; $G_1 = \pi(G_0)$. Pokud chceme, aby byl P polynomiální, poskytneme mu na soukromý vstup isomorfismus π. Interakce probíhá následovně:*

Kroky 1), 2) a 3) se opakují dvakrát

1) *P zvolí náhodně* $i \in \{0;1\}$ *;*
   zvolí náhodně $\tau \in S_n$ *;*
   zkonstruuje $H = \tau(G_i)$ *;*
   H odešle ověřovateli V;

2) *V zvolí náhodně* $j \in \{0;1\}$*;*
j odešle P;

3) *P najde* $\sigma \in S_n : H = \sigma(G_j)$*;*
   - $\sigma = \tau$ *, pokud* $i = j$
   - $\sigma = \tau \circ \pi^{\Delta}, \Delta = (-1)^{i+1}$ *jinak*

   σ *odešle V;*

4) *V důkaz P přijme, pokud* $H = \sigma(G_j)$ *v obou kolech.*

Opět budeme ověřovat podmínky Definice 1.10. Efektivita je zřejmá. Úplnost ověříme též jednoduše: Pokud $G_0 \simeq G_1$, pak dokazovatel bude schopen najít všechna požadovaná zobrazení a důkaz bude přijat s pravděpodobností 1.
Korektnost: Pokud $G_0 \not\simeq G_1$, pak graf H je izomorfní nejvýše jednomu z nich (ať je P^* jakýkoliv). Tedy V vybere j takové, že $G_j \not\simeq H$, s pravděpodobností alespoň jedna polovina. Tedy ať P^* pošle libovolné σ, V ho zamítne. Po dvou kolech přijme (špatný vstup) s pravděpodobností maximálně $\frac{1}{4}$, což jsme chtěli dokázat. Tedy Konstrukce 2.8 představuje interaktivní důkazový systém pro GI.
Zbývá ukázat vlastnost nulové znalosti. Opět předpokládáme $G_0 \simeq G_1$; simulátor M^* je schopen napodobit výpočet, pokud $i = j$. K volbě j dochází ve druhém kroku, to má V k dispozici H, které pro něj představuje náhodný prvek množiny s opakováním $\{\tau(G_i); \tau \in S_n\}$ nezávislý na (náhodně zvoleném) i (obdobně jako v důkazu Tvrzení 2.5). Ani teď V nemá výhodu oproti hádání. Tedy M^* v jednom kole nedokončí simulaci ($\perp$) s pravděpodobností $\frac{1}{2}$. Celkem pravděpodobnost úspěchu M^* tvoří $\frac{1}{4}$. Abychom (pro splnění Definice 1.10) zvýšili tuto hodnotu nad jednu polovinu, necháme simulátor opakovat předchozí pokusy celkem třikrát. Pak:

$$\Pr[M_3^*(x) = \perp] = \prod_{i=1}^{3} \Pr[M^*(x) = \perp] = \left(\frac{3}{4}\right)^3 = \frac{27}{64} < \frac{1}{2}$$

a M^* vydá výstup z pokusu, kdy neselhal. Máme požadovaný simulátor. □

Poznámka 2.9: *V předchozích protokolech jsme svým způsobem přehlíželi jeden důležitý fakt. Totiž dovolili jsme (za sebou) opakovat některé kroky (v uvedených příkladech jenom dvakrát) a přitom jsme předpokládali , že se tím nenaruší vlastnost nulové znalosti. Tento fakt však nemáme dokázaný, i když je pravdivý (pro AI ZK, čemuž konstrukce vyhovují, viz Poznámka 1.19). Proto se buď můžeme odkázat na Kapitolu 3, nebo bychom mohli transformovat výše uvedené konstrukce tak, že bychom posílali ne jeden, ale dva nezávisle náhodně zvolené objekty (indexy, grafy).*

2.3 ZK pro NP

Předvedli jsme důkaz s nulovou znalostí pro problém patřící do *NP*, a jak už jsme se zmínili výše, poskytnutím dokazovateli na soukromý vstup svědka (isomorfismu) pro danou dvojici grafů můžou být oba stroje efektivní (skládání permutací je ‚rychlé'). Přirozeně vyvstává otázka, jestli je takových problémů víc. Odpověď je kladná,

dokonce jsme schopni sestrojit důkaz s nulovou znalostí pro každý jazyk z *NP*. Nepůjde to však udělat bez některých omezení. Za prvé: Budeme schopni prezentovat pouze důkaz s výpočetně nulovou znalostí; za druhé: bude třeba využít pojmu kryptografického závazku a k tomu je potřeba předpokládat (např.) existenci jednosměrných funkce.

Definice 2.10: *Jazyk* L *nazveme NP-úplný, pokud* $L \in NP$ *a* $\forall L' \in NP : L' \prec_p L$ *;*

$\prec_p$ *značí polynomiální redukovatelnost, tj.* $\exists f : \Sigma^* \to \Sigma^*$ *počitatelná polynomiálním strojem a splňující* $\forall x \in \Sigma^* : x \in L' \Leftrightarrow f(x) \in L$

Tedy *NP*-úplný jazyk je takový reprezentant třídy, na kterého můžeme ostatní jazyky převést v polynomiálním čase.

Definice 2.11: *Mějme* S *množinu tajemství (představme si řetězce symbolů),* R *dostatečně velkou množinu náhodných řetězců a* C *dostatečně velkou množinu závazků. Funkci* $c : S \times R \to C$ *nazveme závazek pro* $s \in S$ *, pokud platí:*

- *pro každý pravděpodobnostní polynomiální stroj D, každý polynom p(·), každé* $s \in S$ *a náhodně zvolené* $r \in R$ *:*

$$\Pr\left[D\left(c(s,r)\right) = s \right] < \frac{1}{p\left(\left|c(s,r)\right|\right)} \qquad \textit{(podmínka tajnosti)}$$

- *pro každý pravděpodobnostní polynomiální stroj D a každý polynom p(·), pravděpodobnost nalezení pro libovolná* $r_1, r_2 \in R$ *nějakých* $s_1, s_2 \in S$ *takových, že* $s_1 \neq s_2 \wedge c(s_1, r_1) = c(s_2, r_2)$*, je menší než* $\dfrac{1}{p\left(\left|c(s_1, r_1)\right|\right)}$

(podmínka závaznosti)

Definice 2.11 pouze formalizuje přirozené požadavky zavázání se k nějaké tajné informaci – tajnost říká, že žádný efektivní algoritmus není schopen ze závazku spočítat tajemství. Na druhou stranu závaznost zajišťuje (výpočetní) nepopiratelnost informace, ke které se někdo zavazuje.

Zavázání se a potvrzení probíhá tak, že nejprve se strana *A* zaváže k tajemství *s*, tj. zveřejní $c(s,r)$. V tuto chvíli strana *B* pořád není schopna zjistit toto *s* (díky podmínce tajnosti). Ve chvíli, kdy je třeba prokázat, že *A* tajemství znala od začátku, *A* odhalí hodnoty *s*, *r* a strana *B* se může přesvědčit o pravdivosti tohoto tvrzení (díky podmínce závaznosti). Dostatečně velkou náhodnou množinu *R* potřebujeme, abychom zabránili útoku hrubou silou v případě malé velikosti množiny *S* (snahou projít všechna možná *s*).

Teď máme vše potřebné, abychom ukázali, že $NP \subseteq CZK$. Sestrojíme důkaz s (výpočetně) nulovou znalostí pro NP-úplný problém. Pak sestrojení důkazu pro libovolný jazyk z NP bude spočívat v převedení na daný problém.

Definice 2.12*: G3C (graph-3-coloring) je jazyk obsahující všechny grafy obarvitelné třemi barvami. To znamená:*

$$pro \quad G = (V, E) \;\; \exists \Phi : V \to \{1;2;3\} : \;\; \{i, j\} \in E \Rightarrow \Phi(i) \neq \Phi(j) .$$

Konstrukce 2.13: *Potřebujeme sestrojit* $\langle P, V\rangle$ *splňující podmínky CZK . Na společném vstupu máme* $G=(V,E)$, *P mající na soukromém vstupu obarvení* $\Phi: V \to \{1;2;3\}$ *opět může být pravděpodobnostní polynomiální stroj.*

Kroky 1), 2), 3) a 4) se opakují $2|E|$ *-krát*

1) *P zvolí náhodně permutaci* π *na množině* $\{1;2;3\}$*;*
   dále vytvoří závazky $c_i = c(\pi(\Phi(i)), r_i)\ \forall i \in V$ *pro náhodně zvolená* $r_i \in R$*;*
   $(c_1,\ldots,c_n)$ *odešle ověřovateli V;*
2) *V zvolí náhodně hranu* $\{k,l\} \in E$*;*
   $\{k,l\}$ *odešle P;*
3) *P odešle* $(\pi(\Phi(k)), r_k)$ *a* $(\pi(\Phi(l)), r_l)$*;*
4) *V zkontroluje* $c_k = c(\pi(\Phi(k)), r_k)$, $c_l = c(\pi(\Phi(l)), r_l)$ *a* $\pi(\Phi(k)) \neq \pi(\Phi(l))$*;*
5) *V přijme, pokud 4) platí pro všechna opakování;*

Věta 2.13: *G3C* $\in$ *CZK*

Idea důkazu: Efektivita a úplnost je zřejmá (při dodržení Konstrukce 2.13 je správný vstup přijat s pravděpodobností 1). Korektnost plyne z toho, že pro libovolné *P** a neobarvitelný graf je závazek chybný pro alespoň jednu hranu - toto *V* odhalí s pravděpodobností $(|E|)^{-1}$. Dostatečným počtem opakování pravděpodobnost chybného přijetí snížíme pod $\frac{1}{3}$. Zbývá ukázat nulovou znalost, tj. sestrojit simulátor *M* - ten funguje následovně:

1') zvolí náhodně hranu $\{i', j'\} \in E$, dvojici různých barev $(\chi_{i'}, \chi_{j'}) \in \{1; 2; 3\}^2$ a $r_{i'}, r_{j'} \in R$;
   vytvoří závazky $c_{i'} = c(\chi_{i'}, r_{i'})$ a $c_{j'} = c(\chi_{j'}, r_{j'})$, ostatní závazky zvolí libovolně (c_k pro $k \in V \setminus \{i', j'\}$);
2') zvolí náhodně hranu $\{i, j\} \in E$;
3') $\{i, j\} \neq \{i', j'\} \Rightarrow$ simulátor selže (vydá $\perp$)
   $\{i, j\} = \{i', j'\} \Rightarrow$ pošle klíče k závazkům ($(\chi_{i'}, r_{i'})$ a $(\chi_{j'}, r_{j'})$);
4') dokončí simulaci;

Úspěšná simulace nastane s pravděpodobností $(|E|)^{-1}$. Polynomiálním opakováním bychom zajistili počet úspěchů aspoň v polovině případů (přes Chernoffovu nerovnost). Poté zbývá dokázat, že soubory náhodných veličin představující závazky ve skutečné komunikaci (přípustná obarvení) a při simulaci (z větší části náhodné hodnoty) jsou výpočetně nerozlišitelné. Toto by vyplynulo z definice závazku. Podrobnosti viz [3] □

Poznámka 2.14: *Pro každý jazyk* $L \in NP$ *tedy vytvoření ZK protokolu by probíhalo tak, že bychom použili nějakou ze standardních redukcí na G3C (např. Karpovu redukci), pro který už máme sestrojený důkaz s nulovou znalostí. Ještě by bylo potřeba*

ukázat, že polynomiální redukcí převádíme zároveň i svědka L na svědka G3C (tj. obarvitelnost) a hlavně že redukcí se zachová vlastnost nulové znalosti. Poslední podmínka plyne z toho, že Konstrukce 2.13 splňuje podmínky AI ZK, tj. ověřovateli dovolujeme pomocný vstup (právě vstup z L, který potom zredukujeme na graf pro G3C), protože simulátor přistupuje k ověřovateli jako k černé skříňce a tedy použijeme Poznámku 1.19.

Kapitola 3

Skládání protokolů

3.1 Typy skládání a nedosažitelné množiny

Jednou z důležitých otázek tykajících se interaktivních protokolů s vlastností nulové znalosti je, zda se vlastnost ‚nic neprozradit' zachovává, i když budeme provádět vícero komunikací. Přirozeně nastávají tři možnosti pro tuto situaci: buď provádíme protokoly jeden po druhém (protokol můžeme být jen jeden, který opakujeme) – tj. sekvenčně (sériově), nebo komunikace probíhá najednou se všemi účastníky (odesílání *i*-té zprávy proběhne pro všechny strany současně) – tj. paralelně, anebo dochází ke kombinaci předchozího, kdy v každém kroku můžeme komunikovat s více objekty a jednotlivá kroky provádíme opakovaně (přičemž ne každého kroku se nemusí účastnit všechny strany) – tzv. souběžné skládání. Rovnou poznamenejme že se budeme téměř výhradně zabývat sekvenčním a paralelním skládáním. Tato otázka představuje zajímavou oblast ke zkoumání nejen z hlediska teorie, ale i praxe.

Zatímco dodržení (protokolem) předepsaných pravidel znamená mj. nezávislé provádění každého opakování, už jsme se přesvědčili, že nepoctivý účastník se může (a za účelem narušení bezpečnosti bude pokoušet) libovolně odchýlit od pravidel. Naznačme, čím se liší nepoctivý účastník: postup v jednotlivém kroku může být závislý na zprávách, které obdržel v jiných kolech a tedy přinejmenším nemůžeme předpokládat nezávislost v jeho chování, spíše naopak jeho postup bude nějakým způsobem koordinovaný. Tedy intuitivní přístup „když neprozradím nic v jednom kole, neprozradím nic ani ve více kolech" rozhodně potřebuje podložit formálním zkoumáním.

Technická poznámka: V této kapitole přejdeme od obecné abecedy Σ^* k abecedě binární, $\{0;1\}^*$.

Nejprve potřebujeme uvést definici obvodů.

Definice 3.1: *Booleovský obvod je orientovaný acyklický graf, kde vnitřní vrcholy jsou označeny* $\{\wedge,\vee,\neg\}$. *Vrcholům bez vstupních hran se říká vstupní (nejsou označeny žádným symbolem), bez výstupních hran vrcholy výstupní. Vrchol ‚*$\neg$*' má právě jednu vstupní hranu. Výpočet obvodu probíhá tak, že na vstupní vrcholy se vloží vstup (jeden bit na každý vstup) a pokračuje následovně: vrcholu je přiřazena hodnota tak, že složí hodnoty z vrcholů vstupních hran pomocí operace, která je danému vrcholu přiřazena. Výstupem obvodu je posloupnost bitů na výstupních vrcholech. Velikostí obvodu rozumíme počet vrcholů označených* $\{\wedge,\vee,\neg\}$. *Pravděpodobnostními obvody rozumíme obvody, které mají dodatečný náhodný vstup (uniformně volená posloupnost bitů). Posloupností polynomiálně velkých obvodů rozumíme nekonečnou posloupnost Booleovských obvodů* $C_1, C_2, \ldots$ *takovou, že pro každé n má* C_n *právě n vstupních vrcholů a velikost p(n), kde p(·) je polynom fixovaný pro celou posloupnost.*

Definice 3.2: *Jazyk L patří do neuniformní třídy* ***P/poly****, pokud existuje posloupnost polynomiálních obvodů, která ho rozhoduje.*

Poznámka 3.3: *Ekvivalentní definicí třídy* $P/poly$ *je požadavek na existenci (deterministického) polynomiálního stroje T a posloupnosti slov* $\{a_n\}_{n=1}^{\infty}, \forall n : |a_n| \le p(n)$ *pro nějaký polynom* $p(\cdot)$ *takových, že platí:* $\forall x, |x| = n : x \in L \Leftrightarrow T(x, a_n)$. *Jinými slovy: Pro rozhodování L buď můžeme mít nekonečnou posloupnost polynomiálně velkých obvodů, nebo jeden stroj a nekonečnou posloupnost (polynomiálně velkých) pomocných slov. Neuniformita je zachycena použitím pro vstupy různých délek různých objektů (obvodů nebo nápověd).*

Pro vyvrácení některých hypotéz využijeme tzv. (pseudo)náhodných nedosažitelných množin.

Definice 3.4: *Množinu* $S \subseteq \{0;1\}^k$ *nazveme* $(\tau(k), \varepsilon(k))$***-pseudonáhodná****, pokud pro libovolný (pravděpodobnostní) obvod C velikosti* $\tau(k)$ *s k vstupy a jedním výstupem platí:*

$$\left| \Pr\left[C(S) = 1 \right] - \Pr\left[C\left(\{0;1\}^k \right) = 1 \right] \right| \le \varepsilon(k)$$

Slovy - žádný obvod není schopen rozlišit, jestli dostal prvek z množiny S nebo náhodný řetězec délky k. Posloupnost $S_1, S_2, \ldots$ budeme označovat jako pseudonáhodný soubor (náhodných veličin) (protože soubor uniformních distribucí na této posloupnosti $S_1, S_2, \ldots$, kde pro každé k je S_k $(\tau(k), \varepsilon(k))$-pseudonáhodná množina a funkce $\tau(n), \varepsilon^{-1}(n)$ rostou rychleji než jakýkoliv polynom, takový pseudonáhodný soubor tvoří).

Definice 3.5: *Nechť* $S_1, S_2, \ldots$ *je posloupnost (neprázdných) množin takových, že* $S_n \subseteq \{0;1\}^{Q(n)}$ *pro pevný polynom* $Q(\cdot)$. *Takovou posloupnost nazveme* ***polynomiálně nedosažitelný soubor****, ozn. P-nedosažitelný, pokud pro každý pravděpodobnostní polynomiální stroj D, každou konstantu* $c > 0$, *dostatečně velké n a každé* $x \in \{0;1\}^n$ *platí:*

$$\Pr[D(x) \in S_n] < \frac{1}{n^c}$$

Tedy nedosažitelnost znamená, že žádný efektivní algoritmus není schopen najít prvek z množiny (s nezanedbatelnou pravděpodobností). Následující věta říká, že takové soubory existují a jednotlivé množiny jsou sestrojitelné.

Věta 3.6: *Existuje P-nedosažitelný pseudonáhodný soubor* $S_1, S_2, \ldots$ *pro polynom* $Q(n) = 4n$. *Navíc existuje stroj, který na vstupu* 1^n *vydá na výstup množinu* S_n

Důkaz: odkazujeme na [4]

Pro zkoumání paralelního skládání budeme potřebovat silnější pojem nedosažitelnosti.

Definice 3.7: *Nechť* $Q(\cdot)$ *je polynom a pro* $n = 1, 2, \ldots$ *nechť* $\overline{S}^{(n)}$ *je soubor* 2^n *množin* $\left\{S_1^{(n)}, \ldots, S_{2^n}^{(n)}\right\}$, *kde každá* $S_i^{(n)} \subseteq \{0;1\}^{Q(n)}$. *Posloupnost* $\overline{S}^{(1)}, \overline{S}^{(2)}, \ldots$ *nazveme* ***neuniformně polynomiálně nedosažitelný soubor****, ozn.* $P/poly$*-nedosažitelný, pokud pro každé* $c > 0$, *dostatečně velká n a každý (pravděpodobnostní) obvod C velikosti* n^c *(s n vstupy a* $Q(n)$ *výstupy) platí:*

$$\Pr\left[C(i) \in S_i\right] < \frac{1}{n^c},$$

kde pravděpodobnost se bere přes náhodnost C a uniformně zvolené $i \in \left\{1, \ldots, 2^n\right\}$

Tedy posloupnost $\overline{S}^{(1)}, \overline{S}^{(2)}, \ldots$ tvoří $P/poly$-nedosažitelný soubor, pokud jakýkoliv obvod (polynomiální v *n*), který dostane náhodný index jedné z množin v $\overline{S}^{(n)}$, uspěje ve vydání prvku této množiny pouze se zanedbatelnou pravděpodobností. Podobně jako výše uvedeme fakt, že takový soubor existuje.

Věta 3.8: *Existuje* $P/poly$*-nedosažitelný soubor* $\overline{S}^{(1)}, \overline{S}^{(2)}, \ldots$ *pro polynom* $Q(n) = 4n$ *takový, že pro každé n je* $S_i^{(n)}$ ($2^{\frac{n}{4}}, 2^{-\frac{n}{4}}$)*-pseudonáhodná množina velikosti* 2^n. *Navíc existuje stroj, který na vstupu* 1^n *vydá na výstup soubor* $\overline{S}^{(n)}$.

Důkaz: odkazujeme na [5]

Teď máme připraveny všechny pojmy, abychom se mohli věnovat skládání protokolů.

3.2 Sekvenční skládání

Jak už jsme se zmínili výše, sekvenční skládání je situace, kdy jeden (nebo více různých) protokolů se opakují tak, že každý další je spuštěn po tom, co předchozí skončil. Přirozeně bychom chtěli, aby byla zachována vlastnost důkazu s nulovou znalostí, jinak bychom nemohli opakovat už jednou provedený důkaz. Vytváření důkazů s nulovou znalostí se vůbec obvykle provádí tak, že se sestrojí atomární postup zajišťující nezanedbatelný rozdíl mezi mezemi úplnosti a korektnosti a jeho opakováním se meze zvýší na požadovanou úroveň.

Je zajímavé, že pokud uvažujeme ‚základní' definici nulové znalosti, tj. ITM stroje nemají žádný soukromý vstup, pak sekvenční skládání nemusí zachovávat vlastnost nulové znalosti.

Tvrzení 3.9: *Důkazy s výpočetně nulovou znalostí (bez pomocného vstupu) nezachovávají vlastnost nulové znalosti při sekvenčním skládání.*

Důkaz (dle [5]): Mějme P-nedosažitelný pseudonáhodný soubor $S_1, S_2, \ldots$ s polynomem $Q(n) = 4n$ (takový existuje podle Věty 3.6). Dále nechť K je Booleovská funkce ($K : \{0;1\}^* \to \{0;1\}$) těžká ve smyslu $L_K = \{x : K(x) = 1\} \notin BPP$ (to znamená, že pravděpodobnostním polynomiálním strojem neurčíme výstup pro daný vstup). Sestrojíme interaktivní důkazový systém $\langle P, V \rangle$ pro jazyk $L = \{0;1\}^*$ (samozřejmě existuje triviální systém, kdy ověřovatel vždy přijímá, ale pro naše potřeby půjdeme

jinou cestou). Nechť x je na společném vstupu $\langle P,V\rangle$ a $n=|x|$. Protokol bude vypadat následovně:

1) V zvolí náhodně $s\in\{0;1\}^{4n}$;
   s odešle P;
2) $s\in S_n \Rightarrow P$ odešle V hodnotu $K(x)$
   $s\notin S_n \Rightarrow P$ odešle V náhodné $s_0\in S_n$;
3) V vždy přijme (důkaz, že $x\in L$);

Zřejmě $\langle P,V\rangle$ opravdu tvoří interaktivní důkazový systém (vždy platí $x\in L$). Zároveň jde o důkaz nulové znalosti: simulátor M^* musí být schopen simulovat zprávu od dokazovatele P. Situace, kdy ověřovatel pošle $s\in S_n$ může nastat pouze se zanedbatelnou pravděpodobností (pro jakékoli x), protože ověřovatel je (libovolný) pravděpodobnostní polynomiální stroj a S_n je P-nedosažitelná množina; tedy simulátor se může vždy chovat jako kdyby $s\notin S_n$. V tomto případě musí poslat $s_0\in S_n$. Místo toho simulátor pošle náhodné $s_0\in\{0;1\}^{4n}$. Nicméně tato volba bude výpočetně neodlišitelná od případu $s_0\in S_n$, protože S_n je pseudonáhodná podmnožina $\{0;1\}^{4n}$ a P vybírá $s_0\in S_n$ náhodně (uniformně).
Rozebereme případ, kdy bychom prováděli protokol dvakrát za sebou. Zdůrazněme, že pro obě opakování využíváme stejný P-nedosažitelný soubor, který nezávisí na vstupu x, ale pouze na jeho délce. Toto dovoluje nepoctivému ověřovateli se v druhém kole odchýlit a využít $s_0\in S_n$ z prvního kola (které obdrží s pravděpodobností zanedbatelně blízkou 1). V druhém kole pak V^* nevolí náhodné $s\in\{0;1\}^{4n}$, ale využije s_0. Tím od P získá $K(x')$ (kde x' je vstup pro druhé kolo; může platit $x'=x$). Takový postup není možné (pravděpodobnostním polynomiálním) simulátorem M^* napodobit – z předpokladu M^* neumí $K(x)$ spočítat. Přitom na vstupu (x, x') stroj P po interakci s V^* pošle hodnotu $K(x)$ s pravděpodobností zanedbatelně blízkou 1. Tím pádem dvojkolový protokol není důkazem s výpočetně nulovou znalostí. □

Ač se zdá být výše uvedený příklad velice umělým, ve skutečnosti by šel upravit pro libovolný jazyk ze ZK. Zároveň důkaz Tvrzení 3.9 názorně ilustruje, že bychom potřebovali zesílit nulovou znalost vzhledem k pomocnému vstupu, který by mohl (podvádějící) ověřovatel mít. Pak opravdu je možné sekvenčně skládat protokoly. Připomeňme si, že vlastnost nulové znalosti je vlastností dokazovatele, která má platit pouze pro vstupy z jazyka. Nakonec pojem ‚s pomocným vstupem' znamená, že musíme brát v potaz všechny možné pomocné vstupy.

Tvrzení 3.10: *Mějme ITM P dokazovatele, který splňuje podmínku nulové znalosti vzhledem k pomocnému vstupu pro určitý jazyk L. Mějme polynom $Q(\cdot)$ a nechť P_Q je ITM takový, že na společném vstupu x provádí $Q(|x|)$ opakování, kde každé z nich sestává z volání P na společném vstupu x. (Pro P pravděpodobnostní, P_Q v každém opakování volí náhodné bity nezávisle.) Potom P_Q také splňuje podmínku nulové znalosti (vzhledem k pomocnému vstupu). Navíc, pokud P byl AI PZK dokazovatel, je takový i P_Q.*

Důkaz (dle návodu v [3])*:* Naším úkolem je ukázat, že pro (libovolného) ověřovatele V^* komunikujícího s P_Q jsme schopni sestrojit simulátor M^* napodobující jejich interakci. Hlavní myšlenkou důkazu je simulovat tuto komunikaci v $Q(|x|)$ fázích, kde v každé fázi použijeme simulátor existující pro atomárního dokazovatele P. Všechno, co se V^* dozví v jednotlivých fázích je předáno dál přes jeho pomocný vstup. Musíme tedy rozdělit celý protokol pro libovolný stroj V^* na části, což můžeme učinit, protože poctivý dokazovatel P_Q volá P, který provede svou předepsanou komunikaci a ukončí dílčí interakci (např. zvláštním symbolem). Takže místo V^* lze uvažovat V^{**}, který provádí $Q(|x|)$ postupných interakcí s P. Formálně:

Lemma 3.10.1: *Existuje pravděpodobnostní polynomiální stroj V** takový, že pro každý veřejný a pomocný vstup* (x, z) *platí:*

$$\langle P, V^*(z)\rangle(x) = Z^{(Q(|x|))}, \text{ kde } Z^{(0)} = z;\ Z^{(i+1)} = \langle P, V^{**}(Z^{(i)})\rangle(x), i = 0, \ldots, Q(x) - 1$$

Tj. probíhá postupná interakce V^{**} s P, kde každý další pomocný vstup je výstup předchozí komunikace, a konečný výsledek, tj. náhodná proměnná po $Q(|x|)$ fázích, je stejný jako kdyby proběhla komunikace V^* s P.

Důkaz: Stroj V^{**} prostě napodobuje práci stroje V^* sledováním, jak se mění jeho pracovní (a komunikační) pásky během jednotlivých fází. Mírně upravíme V^* tak, že si nejprve zkopíruje veřejný, soukromý a náhodný vstupy na pracovní pásku a už bude pracovat jen s ní. Stejně tak na začátku každé fáze zkopíruje příchozí zprávu na pracovní pásku. (I těmito úpravami bychom formálně měli přejít k jinému stroji W^*, ale budeme dál psát V^*.) Tedy komunikace $V^*(z)$ s P na společné vstupu x probíhá v $Q(|x|)$ fázích, kdy (až na začátek interakce) V^* nepřistupuje ke svým vstupním páskám a v každé jednotlivé fázi nečte předešlé zprávy ze vstupně-komunikační pásky (může však tyto staré zprávy číst z ‚archivu' na pracovní pásce). Sledováním pracovních pásek na konci každé fáze teď můžeme přejít k vytvoření V^{**}: nechť x je veřejný a z' pomocný vstup; V^{**} zkopíruje z' na pracovní pásku V^* a spustí jeho program pro jednu fázi interakce s P_Q, přičemž V^* považuje komunikační pásky V^{**} za své. Až V^* skončí, stroj V^{**} vypíše pracovní pásku stroje V^* na výstup. Tímto způsobem, pokud z' představuje možný obsah pracovní pásky V^* po $i \geq 1$ fázích, emulace V^* probíhá jako výpočet stroje V^* v $(i+1)$-ní fázi a výstup V^{**} po skončení je distribuován jako obsah pracovní pásky V^* po $(i+1)$-ní fázi. Pro $i = 0$ situace se liší tím, že stroj V^* má pracovní pásku prázdnou, a považuje vstupní pásky (veřejnou, náhodnou a pomocnou) V^{**} za své. Pak platí tvrzení lemmatu. ∟

Rozdělili jsme tedy provádění celkového protokolu na fáze, ve kterých probíhá komunikace (atomárního) dokazovatele P a pravděpodobnostního polynomiálního ověřovatele (s pomocným vstupem) V^{**}, konečný výstup kterého po $Q(|x|)$ fázích odpovídá výstupu V^* po komunikaci s P_Q. Podle předpokladů existuje simulátor polynomiální v prvním vstupu napodobující komunikaci P a V^{**} – označme ho M^{**}. Pro případ perfektní nulové znalosti je mu dovoleno s pravděpodobností jedné poloviny nevydat žádný výstup; opakováním simulace lze tuto pravděpodobnost exponenciálně snížit, proto pro jednoduchost budeme považovat simulaci za úspěšnou. Pak pro každý pravděpodobnostní polynomiální (v x) stroj D, každý polynom $p(\cdot)$, každé dostatečně velké $x \in L$ a každé $z \in \{0;1\}^*$ platí:

$$\left|\Pr\left[D\left(x,z,\left\langle P,V^{**}(z)\right\rangle(x)\right)=1\right]-\Pr\left[D\left(x,z,M^{**}(x,z)\right)=1\right]\right|<\frac{1}{p(|x|)}$$

Teď zkonstruujeme simulátor M^*, který napodobuje komunikaci V^* s P_Q. Bez újmy na obecnosti předpokládejme, že výstup V^* se rovná obsahu jeho pracovní pásky (z obsahu pásky lze výstup sestrojit v polynomiálním čase). Stroj M^* bude využívat simulátor M^{**} (jako černou skříňku).

Simulátor M^* – pracuje induktivně v $Q(|x|)$ fázích; vstupem je (x, z)

0. fáze: nastaví $z^{(0)} = z$;

i-tá fáze: $z^{(i)} = M^{**}(x, z^{(i-1)})$;

Po $Q(|x|)$-té fázi stroj M^* vypíše na výstup $z^{(Q(|x|))}$.

M^* zřejmě pracuje v polynomiálním (vzhledem k prvnímu vstupu) čase. Zbývá ukázat, že M^* produkuje výstup výpočetně nerozlišitelný od výstupu V^* po interakci s P_Q.

Lemma 3.10.2: *Pro každý pravděpodobnostní stroj D polynomiální vzhledem ke svému prvnímu vstupu, každý polynom p(·), každé dostatečně velké* $x \in L$ *a každé* $z \in \{0;1\}^*$ *platí:*

$$\left|\Pr\left[D\left(x,z,\left\langle P_Q,V^*(z)\right\rangle(x)\right)=1\right]-\Pr\left[D\left(x,z,M^*(x,z)\right)=1\right]\right|<\frac{1}{p\left(|x|\right)}$$

Navíc, pokud P splňuje vlastnost perfektní nulové znalosti, pak $\left\langle P_Q,V^*(z)\right\rangle(x)$ *a* $M^*(x,z)$ *jsou identicky rozdělené.*

Urovnejme pojmy: Protokol, kterého se účastní dokazovatel P_Q, ověřovatel V^* a jeho simulátor M^*, je rozdělen na $Q(|x|)$ fází, ve kterých pracují atomární dokazovatel P, ověřovatel V^{**} a jeho simulátor M^{**}. O poslední trojici (dolně-úrovňové, dvojhvězdičkové) víme, že tvoří důkaz s výpočetně (dokonale) nulovou znalostí vzhledem k pomocnému vstupu.

Důkaz: Použijeme tzv. hybridní argument (v podstatě trojúhelníková nerovnost). Zadefinujeme si následující hybridy distribucí pravděpodobnosti: i-tý hybrid, pro $0 \le i \le Q(|x|)$, odpovídá náhodnému procesu, kdy nejprve necháme interagovat V^{**} s P po dobu i fází (začínáme s veřejným vstupem x a pomocným vstupem z; $Z^{(i)}$ značí výstup V^{**} po i-té fázi); ve zbylých $Q(|x|)-i$ fázích provádíme simulaci strojem M^{**}. V obou případech používáme výstup předchozí fáze jako pomocný vstup následující. Formálně hybrid $H^{(i)}$:

$$H^{(i)}(x,z) = M^{**}_{Q(|x|)-i}(x, Z^{(i)}), \text{ kde } Z^{(i)} \text{ jsou jako v Lemmatu 3.10.1 a}$$
$$M^{**}_0(x,z') = z';\ M^{**}_k(x,z') = M^{**}_{k-1}(x, M^{**}(x,z')),\ k = 1,\ldots,Q(x)-i$$

Vidíme, že (dle Lemmatu 3.10.1) $H^{(Q(|x|))}(x,z) = Z^{(Q(|x|))} = \left\langle P_Q, V^*(z)\right\rangle(x)$ a na druhou stranu (podle způsobu konstrukce M^*) $H^{(0)}(x,z) = M^{**}_{Q(|x|)}(x,z) = M^*(x,z)$. Tedy extremální hybridy tvoří přesně distribuce, které nás zajímají. Pro dokázání lemmatu stačí ukázat, že jakékoli dva sousedící hybridy jsou výpočetně nerozlišitelné (protože

výpočetní nerozlišitelnost se tranzitivně přenáší přes polynomiální počet souborů – plyne z trojúhelníkové nerovnosti). Podíváme se na dva sousedící hybridy:

$$H^{(i)}(x,z) = M^{**}_{Q(|x|)-i}(x, Z^{(i)}) = M^{**}_{Q(|x|)-i}\left(x, \left\langle P, V^{**}(Z^{(i-1)})\right\rangle(x)\right)$$

$$H^{(i-1)}(x,z) = M^{**}_{Q(|x|)-(i-1)}(x, Z^{(i-1)}) = M^{**}_{Q(|x|)-i}\left(x, M^{**}(x, Z^{(i-1)})\right)$$

($Z^{(i-1)}$ opět jako v Lemmatu 3.10.1)

Pro spor předpokládejme, že existuje pravděpodobnostní polynomiální stroj D a polynom p(·) tak, že pro nekonečně mnoho $x \in L$ existuje $z \in \{0;1\}^*$ tak, že platí:

$$\left|\Pr\left[D\left(x,z,H^{(i)}(x,z)\right)=1\right] - \Pr\left[D\left(x,z,H^{(i-1)}(x,z)\right)=1\right]\right| > \varepsilon(|x|) = \frac{1}{p(|x|)}$$

Hybridy se v prvních *(i-1)* krocích shodují (komunikace P a V^{**} na společném x a pomocném z), tedy odlišit dvě distribuce lze až od i-tého kroku a pravděpodobnost odlišení lze napsat jako vážený průměr:

$$\Pr\left[D\left(x,z,H^{(i)}(x,z)\right)=1\right] = \sum_h \Pr\left[Z^{(i-1)}=h\right]\cdot\Pr\left[D\left(x,z,M^{**}_{Q(|x|)-i}\left(x,\left\langle P,V^{**}(h)\right\rangle(x)\right)\right)=1\right]$$

$$\Pr\left[D\left(x,z,H^{(i-1)}(x,z)\right)=1\right] = \sum_h \Pr\left[Z^{(i-1)}=h\right]\cdot\Pr\left[D\left(x,z,M^{**}_{Q(|x|)-i}\left(x,M^{**}(x,h)\right)\right)=1\right]$$

Předpokládáme, že stroj D umí hybridy odlišit s nezanedbatelnou pravděpodobností a oba vážené průměry máme nad stejným pravděpodobnostním prostorem, tedy pro každé x, z a i jako výše existuje z' tak, že platí:

$$\left|\Pr\left[D\left(x,z,M^{**}_{Q(|x|)-i}\left(x,\left\langle P,V^{**}(z')\right\rangle(x)\right)\right)=1\right] - \Pr\left[D\left(x,z,M^{**}_{Q(|x|)-i}\left(x,M^{**}(x,z')\right)\right)=1\right]\right| > \varepsilon(|x|)$$

Tímto jsme téměř došli ke sporu – s pomocí algoritmů D a M^{**} vytvoříme nový algoritmus D', který rozlišuje náhodné veličiny $\left(x,z,i,z',\left\langle P,V^{**}(z')\right\rangle(x)\right)$ a $\left(x,z,i,z',M^{**}(x,z')\right)$ a jehož čas běhu D' je polynomiální vzhledem k času D a M^{**}. Pracuje tak, že na vstupu $\left(x,(z,i,z'),\Psi\right)$ (kde Ψ je z $\left\langle P,V^{**}(z')\right\rangle(x)$ nebo $M^{**}(x,z')$) spustí program D se vstupem $M^{**}_{Q(|x|)-i}(x,\Psi)$ a vypíše výstup D.

Zřejmě :

$$\left|\Pr\left[D'\left(x,(z,i,z'),\left\langle P,V^{**}(z')\right\rangle(x)\right)=1\right] - \Pr\left[D'\left(x,(z,i,z'),M^{**}(x,z')\right)=1\right]\right| \geq$$

$$\geq \left|\Pr\left[D\left(x,z,H^{(i)}(x,z)\right)=1\right] - \Pr\left[D\left(x,z,H^{(i-1)}(x,z)\right)=1\right]\right|$$

Zbývá vyřešit jediný problém, a to: D' využívá vstupu $\left(x,z,i,z'\right)$ při rozlišování $\left\langle P,V^{**}(z')\right\rangle(x)$ a $M^{**}(x,z')$; podle Definice 1.16 rozlišovací stroj má přístup pouze k $\left(x,z'\right)$ (a řetězci, jež má rozlišit). Jinými slovy jsme sestavili neuniformní rozlišovací stroj $D' = D'_{i,z}$ závislý na i a z. V případě perfektní nulové znalosti by ale neměly být identické soubory rozlišitelné žádnou funkcí (nejenom efektivním algoritmem), tedy spor máme z pouhé existence. Pro výpočetní nulovou znalost by (dle Poznámky 1.17 a

Lemmatu 3.10.3) neměly být dané soubory odlišitelné ani s pomocí neuniformních (polynomiálně velkých) odlišovatelů, tj. posloupnosti obvodů, kam $D' = D'_{i,z}$ spadá. Tedy jsme došli ke sporu s předpokladem, že atomární dokazovatel *P* splňuje podmínku nulové znalosti (tj. existuje simulátor *M**) . ∟

Posledním krokem je ono zmíněné lemma o neodlišitelnosti ani s pomocí obvodů.

Lemma 3.10.3: *Výstup simulátoru M* z Definice 1.16 není od skutečné interakce V* s P odlišitelný ani s pomocí posloupnosti obvodů, tj. za předpokladů Definice 1.16 pro každou posloupnost polynomiálně velkých obvodů* $C_1, C_2, \ldots$, *každý polynom* $p(\cdot)$, *všechna dostatečně velká n, všechna* $x \in L \cap \{0;1\}^*$, $y \in P_L(x)$ *a* $z \in \{0;1\}^*$ *platí:*

$$\left|\Pr\left[C_n\left(x, z, \langle P(y), V^*(z)\rangle(x)\right) = 1\right] - \Pr\left[C_n\left(x, z, M^*(x, z)\right) = 1\right]\right| < \frac{1}{p(|x|)}$$

Důkaz: Sporem – nechť existuje posloupnost polynomiálně velkých obvodů $\{C_n\}_{n \in N}$ taková, že pro nekonečně mnoho *n* existují trojice (x, y, z) tak, že C_n s nezanedbatelnou pravděpodobností odlišuje výše uvedené soubory. Nechť z' je pomocný vstup vytvořený z původního pomocného vstupu *z* složeného s popisem obvodu C_n následujícím způsobem: nechť $q(\cdot)$ je polynom omezující časovou složitost *V** i *M**. Bez újmy na obecnosti lze předpokládat $|z| \leq q(n)$ (zbytek pomocného vstupu *z* by stroje *V**, *M** nestihli přečíst). Pak $z' = (z, (q(n) - |z|) \times \delta, c_n)$, kde δ značí (pseudo)prázdný symbol na pásce a c_n popis obvodu C_n. Tedy pomocný vstup natáhneme prázdnými symboly na hranici zpracovatelnosti stroji *V** a *M**, a za tu přidáme popis obvodu C_n. Všimněme si: $M^*(x, z') = M^*(x, z)$, $\langle P(y), V^*(z')\rangle(x) = \langle P(y), V^*(z)\rangle(x)$. Vzhledem k tomu, že odlišovatel *D* se vytváří až po tom, co je zafixováno polynomiální omezení *M**, a délka popisu obvodu C_n je omezena polynomem, můžeme sestrojit pravděpodobnostní polynomiální stroj *D,* který je schopen si přečíst popis obvodu C_n a napodobit jeho výpočet, tudíž $D(x, z', \alpha) = C_n(x, z, \alpha)$ (kde α je jeden ze souborů, který máme odlišit). A tím jsme došli ke sporu, že *M** produkuje výpočetně neodlišitelný výstup. ∟

Lemma 3.10.4: *Nechť jsou* $\{R_x\}_{x \in L}, \{S_x\}_{x \in L}$ *a* $\{T_x\}_{x \in L}$ *soubory náhodných veličin, tak, že následující dvojice jsou výpočetně nerozlišitelné.* $\{R_x\}_{x \in L} \equiv_c \{S_x\}_{x \in L}$ *a* $\{S_x\}_{x \in L} \equiv_c \{T_x\}_{x \in L}$*; pak jsou výpočetně nerozlišitelné i* $\{R_x\}_{x \in L}$ *a* $\{T_x\}_{x \in L}$.

Důkaz: Dle předpokladů pro každý polynomiální stoj *D*, každý polynom $p(\cdot)$, pro dostatečně dlouhé $x \in L$:

$$\left|\Pr[D(x, R_x) = 1] - \Pr[D(x, S_x) = 1]\right| < \frac{1}{p(|x|)}$$

$$\left|\Pr[D(x, S_x) = 1] - \Pr[D(x, T_x) = 1]\right| < \frac{1}{p(|x|)}$$

Sečtením nerovnic:

$$\left|\Pr[D(x,R_x)=1]-\Pr[D(x,S_x)=1]\right|+\left|\Pr[D(x,R_x)=1]-\Pr[D(x,S_x)=1]\right|<\frac{1}{\frac{1}{2}\,p(|x|)}$$

Z trojúhelníkové nerovnosti pro absolutní hodnotu a substituce $q(|x|)=\frac{1}{2}\,p(|x|)$:

$$\frac{1}{q(|x|)}>\left|\Pr[D(x,R_x)=1]-\Pr[D(x,S_x)=1]\right|+\left|\Pr[D(x,S_x)=1]-\Pr[D(x,T_x)=1]\right|\geq$$
$$\geq\left|\Pr[D(x,R_x)=1]-\Pr[D(x,T_x)=1]\right|$$

$q(|x|)$ je polynom, a protože jsme předpokládali platnost pro libovolný polynom $p(|x|)$, bude i $\{R_x\}_{x\in L}$ a $\{T_x\}_{x\in L}$ výpočetně nerozlišitelné pro dostatečně velká *x*. ∟

Shrňme důkaz: Rozdělili jsme protokol na jednotlivé atomární fáze, aniž bychom změnili výsledný výstup (Lemma 3.10.1). Definováním tzv. hybridů, jsme ukázali, že výstup námi sestrojeného simulátoru *M** se shoduje s výstupem dokazovatele P_Q a ověřovatele *V** (Lemma 3.10.2) – k tomu jsme došli sporem přes předvedení neuniformního odlišovatele (Lemma 3.10.3) a odlišení hybridů jsme přenesli na odlišení výstupů *M** a *V**, P_Q (Lemma 3.10.4). □

Vidíme tedy, že zesílení definice má svoje opodstatnění a je klíčovým prvkem při vytváření ZK protokolů. Poznamenejme, že většina prakticky konstruovaných důkazů s nulovou znalostí vyhovuje této definici (protože mj. jejich simulátory jsou typu black-box).

3.3 Paralelní skládání

Paralelní skládání probíhá tak, že provádíme více protokolů současně. Jak jsme už popsali výše, účastník provede akce pro všechny protokoly během své aktivní fáze, a pak současně odešle všechny zprávy. Je jasné, že pokud nebyla definice ZK bez pomocného vstupu odolná vůči sekvenčnímu skládání, pak tím spíše nemůže projít paralelní verzí. Důvod, proč by nemělo paralelní skládání fungovat, je ten, že zatímco u sekvenčního postupu jsme mohli znalost střádat a v nějakém kroku ještě neukončeného protokolu ji využít, v paralelním bychom mohli využít dokazovatele v jednom protokolu odpovědět na naše dotazy, které bychom potřebovali do protokolu jiného. Bohužel se ukáže, že ani zesílená definice, tj. AI ZK, neodolá paralelizaci.

Tvrzení 3.11 : *AI ZK není uzavřené na paralelní skládání.*

Důkaz (dle [5]): Předvedeme dva interaktivní důkazové systémy $\langle P_1,V_1\rangle$ a $\langle P_2,V_2\rangle$ takové, že každý splňuje vlastnost nulové znalosti, ale jejich paralelizace nějakou znalost prozradí. Jsou vytvářeny s důrazem na jednoduchost, proto dovolíme prázdné kroky (kdy stroj nic neudělá a jen zpátky přepíše stavový bit). Konstrukce vypadá následovně:
Mějme jazyk *L* a těžkou Booleovskou funkci *K* jako v důkazu Tvrzení 3.9. Na společném vstupu (pro obě dvojice) je *x*, *n* = |*x*|, a obě dvojice využívají *stejný* $P/poly$-nedosažitelný soubor $\overline{S}^{(1)},\overline{S}^{(2)},\ldots$ s $Q(n)=4n$ (takový existuje podle Věty 3.8).

$\langle P_1, V_1 \rangle$:

1) P_1 zvolí náhodně $i \in \{1,\ldots,2^n\}$ a pošle ho V_1;
2) prázdný krok;
3) prázdný krok;
4) V_1 zvolí náhodně $s \in \{0;1\}^{4n}$ a pošle ho P_1;
5) $s \in S_i^{(n)} \Rightarrow$ P_1 odešle V_1 hodnotu $K(x)$
   jinak P_1 nepošle nic (prázdnou zprávu);

$\langle P_2, V_2 \rangle$:

(protokol využívá stejný $P/poly$-nedosažitelný soubor jako předchozí dvojice)

1) prázdný krok ;
2) V_2 zvolí náhodně $j \in \{1,\ldots,2^n\}$ a pošle ho P_2 ;
3) P_2 zvolí náhodně $r \in S_j^{(n)} \Rightarrow$ a pošle ho V_2 ;
4) prázdný krok;
5) prázdný krok;

Ukažme, že oba protokoly jsou AI ZK:

Simulátor pro $\langle P_1, V_1 \rangle$ v prvním kroku může taky vybrat náhodné $i \in \{1,\ldots,2^n\}$; v kroku 5) P_1 posílá $K(x)$, pokud od ověřovatele dostane $s \in S_i^{(n)}$, toto však nastane pouze se zanedbatelnou pravděpodobností, protože (libovolný) ověřovatel je pravděpodobnostní polynomiální stroj (popř. s pomocným vstupem) a pokud by uměl najít nějaké $s' \in S_i^{(n)}$, mohli bychom jej transformovat na obvod C', který by byl ve sporu s podmínkou $P/poly$-nedosažitelnosti souboru $\overline{S}^{(1)}, \overline{S}^{(2)}, \ldots$ (na rozdíl od Definice 3.7, kde má obvod pouze index množiny, jejíž prvky hledá, zde má ještě vstup x; ten mu ale nemůže pomoci, jinak bychom opět byli schopni sestrojit obvod narušující vlastnost nedosažitelnosti, který by měl x ‚zabudované' v sobě). Tedy simulátor se může chovat jakoby od ověřovatele dostal $s \notin S_i^{(n)}$.

Simulátor pro $\langle P_2, V_2 \rangle$ se musí vypořádat pouze s krokem 3), kdy P_2 posílá náhodné $r \in S_j^{(n)}$, tedy pošle náhodné $r' \in \{0;1\}^{4n}$, které bude díky pseudonáhodnosti $S_j^{(n)}$ výpočetně neodlišitelné od dokazovatelova r.

Zbývá předvést dokazovatele V^*, který při paralelním provádění obou protokolů získá nějakou znalost:

Po prvním kroku V^* má náhodné $i \in \{1,\ldots,2^n\}$ od (části) dokazovatele P_1 (část P_2 žádnou zprávu neposílá). V druhém kroku položí $j = i$ (místo aby j volil náhodně). Po 3. kroku od P_1 obdrží $r \in S_i^{(n)}$ a to záhy použije v kroku 4) místo náhodného s. Dokazovatel ověří, že opravdu platí $r \in S_i^{(n)}$ a pošle hodnotu $K(x)$. Tímto V^* obdržel jemu nepřístupnou znalost s pravděpodobností 1 a není možné tuto komunikaci nasimulovat žádným M^*, protože funkce K je těžká (tj. $L_K = \{x : K(x) = 1\} \notin BPP$). □

Předchozí tvrzení není až zas tak špatná zpráva. Sice paralelní skládání by se dalo využít ke snížení počtu kol v důkazech s nulovou znalostí (i když pořád narůstá celková velikost zpráv), na druhou stranu paralelní procesy opravdu mohou vykazovat neobvyklé chování, tedy bylo třeba přistupovat opatrně k hypotéze o paralelním

skládání. Hlavně však konstrukce z důkazu Tvrzení 3.11 je na první pohled velmi umělá, až by se dalo říci schválně nepřátelská. Z praktického hlediska je důležitá následující věta:

Věta 3.12: *Pro každou množinu z NP existuje důkaz s nulovou znalostí uzavřený na paralelní skládání, to vše za podmínky existence těžkých problémů. Navíc je příslušný důkaz proveditelný v konstantním počtu kol.*

Nejprve se podívejme na konstantní počet kol. Toho můžeme dosáhnout právě paralelním skládáním, pokud není potřeba provádět paralelních větví ‚příliš mnoho' (nereálný př. – pokud bychom potřebovali exponenciálně mnoho větví, takový protokol by vůbec nebyl proveditelný, natož ZK, vzhledem k polynomiálnímu omezení ověřovatele).

Větu dokazovat nebudeme, odkážeme se zde na [2], kde je ukázáno, že tzv. Goldreichův-Kahanův protokol pro (nám už známý problém) G3C je uzavřený na paralelní skládání. Totiž lze sestavit simulátor, který je schopen simulovat i paralelní komunikaci s libovolným ověřovatelem. Nicméně se předpokládá, že existují zavazovací schémata se specifickými vlastnostmi (tato část právě vychází z existence claw-free permutací), a navíc simulátor běží v polynomiálním čase v průměru (podobně jako v ZK protokolu pro *GNI*).

Závěr

V této práci jsme se pokusili sestavit jistý přehled o pojmu důkaz s nulovou znalostí a jeho vlastnostech. Vzhledem k zaměření se na sekvenční a paralelní skládání bylo potřeba nejprve porovnat různé definice, které byly v historickém horizontu představeny právě pro zkoumání daných podmínek a všechny se opravdu využívají při prezentaci jednotlivých aspektů nulové znalosti. Kromě několika základních příkladů se právě podíváme na vztah definic a možnosti jednotlivé ZK důkazy skládat výše uvedenými způsoby. Poznatky ohledně těchto dvou druhů opakování jsou zkoumány v různé literatuře z daného oboru, přesto se většinou autoři zaměřují na úžeji vymezené případy, které zkoumají velice podrobně do hloubky. My se snažíme podat přehled o základních faktech a jejich souvislostech, které by pomohly začínajícím zájemcům pomoci se zorientovat a i nasměrovat je případně v dalším zkoumání tématu.

Literatura

[1] Barak B.: *How to go beyond the black-box simulation barrier*, In Proc. 42nd FOCS, pages 106-115. IEEE, 2001

[2] Goldreich O.: *Concurrent Zero-Knowledge with Timing, Revisited*, In 34th ACM Symposium on Theory of Computing, pages 332-340, 2002

[3] Goldreich O.: *Foundations of Cryptography Volume I – Basic Tools*, Cambridge University Press, Cambridge, 2003

[4] Goldreich O., Krawczyk H.: *Sparse pseudorandom distributions*, Random Structures & Algorithms, Vol. 3, No. 2, 1992

[5] Goldreich O., Krawczyk H.: *On the Composition of Zero-Knowledge Proof Systems*, SIAM Journal on Computing, Vol. 25, No. 1, February, pages 169–192, 1996

[6] Goldreich O., Micali S., Wigderson A.: *Proofs that Yield Nothing but Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems*, Journal of the ACM, Vol. 38, No. 1, pages 691–729, 1991

[7] Goldreich O., Oren Y.: *Definitions and Properties of Zero-Knowledge Proof Systems*, Journal of Cryptography, Vol. 7, No. 1, pages 1--32, 1994

[8] Holub Š.: *Složitost pro kryptografii*, elektronická skripta, `http://www.karlin.mff.cuni.cz/~holub/texty.htm`

[9] M. Primas: *Důkazy s nulovou znalostí*, bakalářská práce, MFF UK, 2006