Univerzita Karlova v Praze, Filozofická fakulta
Katedra logiky

Natálie Tejkalová

PROBABILISTICKÉ ALGORITMY PRO PRVOČÍSELNOST

Bakalářská práce

Vedoucí práce:
Doc. RNDr. Vítězslav Švejdar, CSc.

2013

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a že jsem uvedla všechny použité prameny a literaturu.

V Praze 11. srpna 2013

Natálie Tejkalová

Poděkování

Ráda bych poděkovala Doc. RNDr. Vítězslavu Švejdarovi, CSc. za vedení práce a za podnětné rady a připomínky.

Abstrakt

Ačkoli v poslední době byla pozornost upřena především na nový deterministický algoritmus pro testování prvočíselnosti AKS, pravděpodobnostní algoritmy zůstávají efektivním nástrojem pro testování prvočíselnosti. Naše práce se věnuje převážně dvěma nejznámějším probabilistickým algoritmům pro testování prvočíselnosti. Podrobně popisuje princip a důkaz správnosti Solovay-Strassenova a Rabin-Millerova algoritmu. Kromě toho se také pokouší dívat na problematiku pravděpodobnostních testů obecněji. Je představena definice probabilistického algoritmu a různé třídy složitosti odpovídající Monte Carlo či Las Vegas algoritmům. Kromě čistě matematické teorie naznačíme i filosofické aspekty, nad kterými je třeba se při používání pravděpodobnostní metody zamýšlet.

Abstract

Attention has been paid mostly to the new deterministic algorithm for primality testing AKS recently. However, probabilistic algorithms remain an efficient tool for primality testing. Our thesis focuses mostly on two most well-known probabilistic algorithms for primality testing. It describes the main idea and gives proofs of correctness of Solovay-Strassen and Rabin-Miller algorithms. Apart from that, it also tries to look at the subject of probabilistic algorithms from a wider perspective. It presents a definition of a probabilistic algorithm and various complexity classes that correspond to Monte Carlo or Las Vegas algorithms. Besides pure mathematical theory, we mention also some philosophical aspects that need to be considered when we decide to use the probabilistic method.

Obsah

1 Úvod **1**

2 Základní poznatky o prvočíslech **3**
2.1 Vybrané poznatky z teorie čísel 3
2.2 Naivní pravděpodobnostní testy prvočíselnosti 6
2.2.1 Pokusné dělení 7
2.2.2 Hledání největšího společného dělitele 7
2.2.3 Fermatův test 7

3 Solovayův-Strassenův test **10**
3.1 Matematické základy 10
3.1.1 Vlastnosti Jacobiho symbolu 11
3.2 Popis algoritmu 17
3.3 Existence dostatečného počtu svědků 19

4 Rabinův-Millerův test **21**
4.1 Popis algoritmu 22
4.2 Existence dostatečného počtu svědků 24
4.3 Deterministická verze testu 28

5 Teorie probabilistických algoritmů **30**
5.1 Intuitivní popis probabilistických algoritmů 30
5.2 Formální zachycení probabilistického algoritmu 31
5.3 Třídy složitosti pro probabilistické algoritmy 34
5.4 Problém náhodnosti 36
5.4.1 Mírně náhodné generátory 38
5.5 Aplikace pravděpodobnostních metod 38

6 Závěr **41**

Literatura **42**

1 Úvod

V této práci se budeme zabývat úlohou PRIMES, která se pro dané přirozené číslo n ptá, zda je n prvočíslo. Tato úloha je zajímavá jak z teoretického, tak z praktického hlediska. V praxi se testování prvočíselnosti používá v kryptografických algoritmech, například známém algoritmu RSA. Takové algoritmy zajišťují bezpečnost mnohých běžně prováděných operací, např. fungování elektronického bankovnictví či digitálního podpisu.

V teoretické rovině dlouho zůstávala nezodpovězená otázka o příslušnosti úlohy PRIMES do třídy úloh, které mají polynomiální algoritmus — třídy P. Narozdíl od mnoha jiných výpočetně složitých úloh totiž o úloze PRIMES nebylo prokázáno, že by patřila k NP-úplným úlohám. Až v roce 2002 publikovali M. Agrawal, N. Kayal a N. Saxena z indické univerzity v Kanpuru článek PRIMES *is in* P, kde se konečně podařilo ukázat příslušnost úlohy PRIMES do třídy P. Z teoretického hlediska to byl důležitý objev, který upoutal pozornost mnoha dalších vědců. Jejich pozdější snahy směřovaly ke zefektivnění celého algoritmu — původně navržený algoritmus pracoval v čase $O((\log n)^{12})$[1], jeho vylepšení snížila složitost až někam k $O((\log n)^6)$. To je oproti „běžným“ polynomiálním algoritmům pořád poměrně velký exponent.

Pro praxi, která od algoritmu vyžaduje především efektivitu, jsou tak i v dnešní době často lepším řešením probabilistické algoritmy. To jsou algoritmy, které potřebují navíc ještě nějaký náhodný parametr, a správnost výsledku je ovlivněna právě tímto náhodným parametrem. Probabilistické algoritmy obecně prokázaly svou užitečnost právě v souvislosti s testováním prvočíselnosti. V této oblasti jsou známy a používány již od konce 70. let a nabízejí libovolně malé riziko chyby.

Právě pravděpodobnostním algoritmům se budeme věnovat i my. V první kapitole shrneme základní definice a věty, převážně z oblasti teorie grup, jejichž znalost by měla velmi usnadnit čtení následujícího textu. Také představíme Fermatův test a vysvětlíme, proč ho nemůžeme považovat za opravdový pravděpodobnostní test na prvočíselnost. Další dvě kapitoly podrobně rozeberou dva nejznámější pravděpodobností testy prvočíselnosti. Nejdříve

[1] $O()$ označuje asymptotickou složitost v nejhorším případě

popíšeme historicky starší Solovayův-Strassenův test, v další kapitole poté novější a o něco efektivnější Rabinův-Millerův test. V obou případech se budeme snažit nejen o srozumitelný popis fungování algoritmu, ale hlavně o podrobné vysvětlení toho, proč takový algoritmus funguje. Závěrečná, o něco obecnější kapitola, se bude věnovat teorii pravděpodobnostních algoritmů. Přestavíme možnou definici pravděpodobnostního algoritmu a speciální třídy složitosti, které jsou s touto teorií svázány. Stručně nastíníme také možné problémy, které s sebou aplikace pravděpodobnostních algoritmů přináší — zde už se od čisté matematiky dostáváme až k hranicím filosofie.

2 Základní poznatky o prvočíslech

V této kapitole shrneme některá známá fakta o prvočíslech, která budeme potřebovat v důkazech v dalších kapitolách. Připomeneme základní poznatky z teorie grup a ujasníme, v jakých strukturách se budeme pohybovat. V některých případech uvedeme použitá tvrzení i s důkazem, často ho ale vynecháme — jde především o to, poskytnout čtenáři základní rámec, se kterým se bude lépe orientovat v dalším textu. Také popíšeme několik jednoduchých (ale nefungujících) nápadů, jak by pravděpodobnostní test na prvočíselnost mohl vypadat. Díky tomu získáme základní představu o tom, co to je pravděpodobnostní algoritmus a co od něj musíme požadovat.

2.1 Vybrané poznatky z teorie čísel

Definice (Grupa)**.** *Grupou* $(G, 0, -, \circ)$ nazveme strukturu s nosičem G, neutrálním prvkem $0 \in G$, unární operací $-$ a binární operací $\circ$, která splňuje:
(i) $\forall x, y, z \in G : (x \circ y) \circ z = x \circ (y \circ z)$ (asociativita),
(ii) $\forall x \in G : x \circ 0 = x$ (neutrální prvek),
(iii) $\forall x \in G : x \circ (-x) = 0$ (opačný prvek).
Pokud navíc
(iv) $\forall x, y \in G : x \circ y = y \circ x$ (komutativita),
mluvíme o komutativní (Abelově) grupě.

Pokud místo $\circ$ píšeme $+$ a myslíme tedy sčítání, mluvíme o aditivní grupě. Pokud místo $\circ$ píšeme znak pro násobení $\cdot$, označujeme danou grupu jako multiplikativní. Právě v multiplikativních grupách se budeme pohybovat my.
Příklad
1. Celá čísla s binární operací sčítání, $-$ jako unární operací a 0 jako neutrálním prvkem jsou grupa.
2. Jednoduchou multiplikativní grupu dostaneme, pokud pro přirozené číslo

n vezmeme všechna přirozená čísla menší než n nesoudělná s n, neutrální prvek bude 1, binární operaci $\cdot$ představuje násobení modulo n a unární operace $^{-1}$ značí inverzní prvek vůči násobení, tedy platí $aa^{-1} = 1$.

Pro počet čísel menších než n nesoudělných s n přitom máme zvláštní označení — udává ho Eulerova funkce. Při znalosti prvočíselného rozkladu můžeme její hodnotu snadno spočítat podle známých vzorců.

Definice (Eulerova funkce)**.** *Eulerova funkce* $\phi(n)$ udává počet čísel menších než n nesoudělných s n.

Podle Eulerovy funkce budeme odpovídající grupu čísel menších než n nesoudělných s n s operací násobení modulo n nazývat Eulerova grupa a značit $\Phi(n)$. V literatuře často pro stejnou strukturu najdeme i označení $(\mathbb{Z}/nZ)^*$ nebo $(\mathbb{Z}_n)^*$. Velikost této grupy je dána přímo hodnotou Eulerovy funkce $\phi(n)$.

Důležitá vlastnost Eulerovy funkce se týká její počitatelnosti. Pokud známe prvočíselný rozklad čísla n, můžeme Eulerovu funkci snadno spočítat pomocí vzorce. Zatím ovšem nebyl objeven žádný efektivní algoritmus, který by totéž zvládl i bez znalosti rozkladu.

V dalších kapitolách několikrát budeme Eulerovu funkci počítat, pojďme tedy představit vzorce na její výpočet. Pro jejich odvození se používá známá vlastnost Eulerovy funkce, kterou vyjádříme v následující větě.

Věta 1. *Když n, m jsou nesoudělná přirozená čísla, pak $\phi(nm) = \phi(n)\phi(m)$.*

Výpočet Eulerovy funkce

$\phi(p) = p - 1$, pokud p je prvočíslo.
$\phi(p^k) = (p-1)p^{k-1}$, pokud p je prvočíslo a $k \leq 1$ přirozený exponent.
Pokud $n = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$, Eulerovu funkci spočítáme jako
$\phi(n) = \prod_{i=1}^{m} (p_i - 1)p_i^{k_i - 1}$.

Zřejmě nejdůležitější poznatek pro testy prvočíselnosti je Malá Fermatova věta. Její znění se ovšem nevztahuje pouze na prvočísla.

Věta 2 (Malá Fermatova věta)**.** *Nechť $(G, 1, \cdot, ^{-1})$ je (konečná) komutativní grupa a m počet jejích prvků. Pak pro každý prvek $a \in G$ platí $a^m = 1$.*

Důkaz. Zvolme $a \in G$. Nechť $b_1, \dots, b_m$ je seznam všech prvků G. Součiny $b_1 a, \dots, b_m a$ jsou navzájem různé a každý takový součin je opět prvkem G. Z toho vyplývá rovnost množin $\{b_1, \dots, b_m\}$ a $\{b_1 a, \dots, b_m a\}$. Musí se tedy rovnat i součiny prvků obou množin: $\prod_{i=1}^{m} b_i = \prod_{i=1}^{m} b_i a$. Tento vztah mohu snadno upravit na $1 = a^m$, což jsme chtěli dokázat. □

Jako důsledek Malé Fermatovy věty můžeme uvést následující tvrzení, které platí speciálně pro grupy $\Phi(p)$, kde p je prvočíslo.

Tvrzení 1. *Když p je prvočíslo a $a \in \Phi(p)$, pak $a^{p-1} = 1 \bmod p$.*

V teorii grup se často používá několik důležitých pojmů. Pojďme krátce některé vlastnosti grup představit.

Definice (Řád grupy a řád prvku)**.** Uvažujme konečnou multiplikativní grupu G. Řekneme, že grupa G má *řád* m, je-li m počet jejích prvků. Řád prvku $g \in G$ je nejmenší číslo různé od 0 takové, že $g^n = 1$.

Řád prvku v grupě má několik vlastností, které budeme v dalším textu často používat.

Lemma 1. *(a) V konečné grupě má každý prvek nějaký řád.*
(b) Když řád a je k, pak $1, a, a^2, \ldots, a^{k-1}$ jsou navzájem různé.
(c) Nechť řád a je k. Pak $a^m = 1 \Leftrightarrow k \mid m$.
(d) Je-li G komutativní grupa s m prvky a prvek $a \in G$ má řád k, pak $k \mid m$.

Důkaz. (a) Plyne z Malé Fermatovy věty: když $a \in G$ a G má řád k, pak určitě $a^k = 1$.
(b) Kdyby nebyly různé, tzn. existovaly by $i, j, i < j$ takové, že $a^i = a^j$, pak můžeme danou rovnost vynásobit inverzním prvkem k a^i. Získáme vztah $1 = a^{j-i}$. To je ve sporu s tím, že řád a je k ($j - i < k$).
(c) Implikace zprava doleva je jasná. Opačná implikace si zaslouží krátké zdůvodnění: nechť tedy $a^m = 1$. Dělme číslo m se zbytkem číslem k: $m = kj + r$, kde $r < k$. Chceme ukázat, že $r = 0$. Platí tyto rovnosti: $1 = a^m = a^{kj+r} = (a^k)^j a^r = a^r$. Kdyby tedy r bylo různé od nuly, měli bychom číslo menší než k o kterém by platilo, že $a^r = 1$, což je opět spor s tím, že k je řád prvku a.
(d) Plyne z Malé Fermatovy věty a bodu (c). □

Definice (Cyklická grupa)**.** Grupa G s m prvky je *cyklická*, jestliže má prvek řádu m. Takový prvek nazýváme *generátor*.

Označení *generátor* má své vysvětlení: jeho postupným umocňováním dostaneme všechny prvky grupy. Pokud grupa není cyklická, vygenerovat z jednoho prvku ji nelze.

Místo o generátoru cyklické grupy $\Phi(p)$ můžeme také mluvit o *primitivním kořenu prvočísla*.

Definice (Primitivní kořen prvočísla)**.** Primitivní kořen prvočísla p je prvek $r \in \Phi(p)$, jehož řád je $p - 1$.

Věta 3. *Každé prvočíslo má primitivní kořen. Jinak řečeno, grupa* $\Phi(p)$ *je cyklická.*

I tato věta se nám v dalších kapitolách bude mnohokrát hodit. Její důkaz je poměrně rozsáhlý a proto ho zde neuvádíme. Dokonce platí i obecnější tvrzení: když p je liché prvočíslo a e kladné celé číslo, grupa $\Phi(p^e)$ je cyklická. Tento zobecněný poznatek později dokážeme a využijeme při důkazu fungování obou popisovaných algoritmů.

Na závěr tohoto oddílu uvedeme ještě jednu důležitou větu. Tato věta stanovuje podmínky, za kterých mohu místo s jedním číslem pracovat s jeho modulární reprezentací.

Definice. Uvažujme $p_1, p_2, \ldots, p_n$ navzájem nesoudělná čísla. Označme zbytek po dělení čísla a číslem p jako $mod(a, p)$. *Modulární reprezentace* čísla a vůči $p_1, p_2, \ldots, p_n$ je n-tice čísel $[mod(a, p_1), mod(a, p_2), \ldots, mod(a, p_n)]$.

S pomocí vhodné modulární reprezentace můžeme někdy jedinou kongruenci rozložit na soustavu a pracovat s jednotlivými členy soustavy.

Věta 4 (Čínská zbytková věta)**.** *Okruh* $\mathbb{Z}_{\prod_{i=1}^{n} p_i}$ *je izomorfní s kartézským součinem* $\mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \times \ldots \times \mathbb{Z}_{p_n}$.

Ekvivalentní formulace Čínské zbytkové věty říká, že číslo menší než $\prod_{i=1}^{n} p_i$ je svou modulární reprezentací určeno jednoznačně. Podrobně: když $p_1, \ldots, p_n$ jsou nesoudělná čísla, a $a_1, \ldots, a_n$ celá čísla, pak soustava rovnic

$$\begin{aligned} x &\equiv a_1 \bmod p_1 \\ x &\equiv a_2 \bmod p_2 \\ &\vdots \\ x &\equiv a_n \bmod p_n \end{aligned}$$

má řešení a všechna její řešení jsou kongruentní modulo $\prod_{i=1}^{n} p_i$.

2.2 Naivní pravděpodobnostní testy prvočíselnosti

Terminologická poznámka
Při popisu testů prvočíselnosti budeme používat dva téměř intuitivně srozumitelné pojmy: svědek složenosti a pseudoprvočíslo (v nějaké bázi). Svědek složenosti je takové číslo, že po provedení testu s tímto vstupem bude ihned jasné, že testované číslo je složené. V opačném případě prohlásíme, že testované číslo je pseudoprvočíslo v dané bázi.

2.2.1 Pokusné dělení

Nejjednodušší, zcela intuitivní možnou verzí „testu prvočíselnosti“, je hledání konkrétního dělitele. Mějme číslo n, o kterém si přejeme zjistit, zda je to prvočíslo. Mohli bychom postupovat takto: náhodně zvolíme číslo a z intervalu $[2, \ldots, n-1]$. Následně provedeme dělení n/a. Pokud je výsledek celočíselný, našli jsme dělitele a můžeme tedy s jistotou uzavřít, že n není prvočíslo. V opačném případě zkoušíme jiné potenciální dělitele a tak dlouho, dokud nenajdeme dělitele nebo dokud nezískáme pocit, že jsme prvočíselnost ověřili dostatečně.

Problémem takového postupu je skutečnost, že dělitelů nemusí být mnoho — vynásobením dvou velkých prvočísel získáme složené číslo, s jehož odhalením by ale měl popsaný algoritmus velkou obtíž. Abychom mohli tvrdit, že číslo je „pravděpodobně prvočíslo“, museli bychom stejně vyzkoušet téměř všechny možné dělitele (samozřejmě stačí uvažovat pouze liché dělitele menší rovno $\sqrt{m}$, ale to není z našeho pohledu nijak podstatné). Od pravděpodobnostního algoritmu naopak požadujeme, aby se pravděpodobnost chyby s rostoucím počtem iterací blížila nule.

2.2.2 Hledání největšího společného dělitele

Místo přímého testování dělitelnosti můžeme místo toho počítat největšího společného dělitele náhodně zvoleného čísla $a \leq m-1$ a čísla m. Pokud největší společný dělitel vyjde větší než 1, máme důkaz, že testované číslo není prvočíslo. Tento postup je stále velmi jednoduchý (největšího společného dělitele spočítáme s pomocí Eukleidova algoritmu v polynomiálním čase), v mnoha případech ale výrazně zvyšuje šanci na úspěch: například číslo 1387 je součinem dvou prvočísel, zatímco čísel a, pro která $gcd(a, m) > 1$, je 90. Přesto není pravděpodobnost zvolení toho správného náhodného čísla v jednom pokusu moc velká: $90/1386 \approx 6,5\%$.

2.2.3 Fermatův test

Poměrně nadějný postup, jak testovat prvočíselnost, nám ukazuje Fermatova věta. Pro každé prvočíslo p a číslo $a < p$ musí podle ní platit $a^{p-1} \equiv 1 \bmod p$. V případě, že pro nějaké číslo n a číslo a s ním nesoudělné vyjde $a^{n-1} \not\equiv 1 \bmod n$, víme, že číslo n není prvočíslo. Pravděpodobnostní test by tedy postupoval takto: mám číslo n a k němu náhodně vygeneruji/zvolím číslo a. Pomocí Eukleidova algoritmu mohu snadno ověřit, zda jsou obě čísla nesoudělná. Pokud ne, nalezený dělitel slouží jako doklad o tom, že n prvočíslo není. Pokud jsou nesoudělná, spočtu $a^{n-1} \bmod n$. Pokud je výsledek různý od

jedné, opět mohu test uzavřít s prohlášením, že n není prvočíslo. V opačném případě mohu zvolit jinou bázi a a celý postup opakovat.

Bohužel se ukázalo, že existují složená čísla, která s pomocí Fermatova testu není možné odhalit. Těmto číslům říkáme Carmichaelova čísla.

Definice (Carmichaelova čísla)**.** Složené číslo n nazýváme *Carmichaelovo číslo*, pokud pro všechna $a < n$, $gcd(a, n) = 1$ platí, že $a^{n-1} \equiv 1 \bmod n$.

Nejmenší existující Carmichaelovo číslo (objevené právě Carmichaelem roku 1910)[1] je $561 = 3 \cdot 11 \cdot 17$. Kdybychom byli schopni Carmichaelova čísla nějak jednoduše rozpoznat, Fermatův test by by stále mohl fungovat jako opravdový pravděpodobnostní algoritmus pro testování prvočíselnosti — platí následující věta. [16]

Věta 5. *Nechť $m > 1$ je složené číslo a existuje číslo a takové, že $gcd(a, m) = 1$ a zároveň $a^{m-1} \not\equiv 1 \bmod m$ (takže a neprojde Fermatovým testem). Pak alespoň polovina možných bází nesplní Fermatův test:*

$$\left|\{a \leq m - 1 : a^{m-1} \not\equiv 1 \bmod m\}\right| \geq m/2.$$

Důkaz. Množinu Z_m čísel menších než m rozdělíme na tři disjunktní části: $A = \{a; a^{m-1} \equiv 1 \bmod m\}$, $B = \{a; gcd(a, m) = 1 \wedge a^{m-1} \not\equiv 1 \bmod m\}$, $C = \{a; gcd(a, m) \neq 1\}$. Množiny A a C jsou disjunktní, protože $a^{m-1} \equiv 1 \bmod m$ implikuje $gcd(a, m) = 1$ a také opačně $gcd(a, m) \neq 1$ implikuje $a^{m-1} \not\equiv 1 \bmod m$ (prvky okruhu $\mathbb{Z}_m$ jsou invertibilní, právě tehdy, když $gcd(a, m) = 1$). Všichni Fermatovi svědci složenosti jsou obsaženi v množinách B a C. Pro důkaz věty je tedy třeba ukázat, že $|A| < |B| + |C|$.

Předpoklad věty říká, že B je neprázdná, tedy existuje $b_0 \in B$. Uvažme množinu $Ab_0 = \{ab_0; a \in A\}$. Tvrdíme, že $Ab_0 \subseteq B$: jelikož $gcd(a, m) = 1$ a $gcd(b_0, m) = 1$, platí také $gcd(ab_0, m) = 1$. Kdyby $ab_0 \in A$, mohli bychom uvažovat takto: $1 \equiv ab_0^{m-1} \bmod m = a^{m-1}b_0^{m-1} \bmod m = b_0^{m-1} \bmod m \not\equiv 1$, čímž dojdeme ke sporu. Také víme, že $|A| = |Ab_0|$. Z předchozí analýzy dostáváme jako závěr $|A| < |B|$.

Nyní můžeme vyjádřit m jako $m = |A| + |B| + |C| \geq |A| + |A| + 1 > 2\,|A|$. Také $|B| + |C| = m - |A| > |m| - \frac{|m|}{2} = \frac{|m|}{2}$. □

Skutečnost je ovšem taková, že Carmichaelova čísla rozpoznat neumíme. Díky [2] navíc víme, že jich je nekonečně mnoho. Fermatův test nemůže být považován za „opravdový“ pravděpodobnostní test prvočíselnosti, ačkoli k němu má z algoritmů zatím popisovaných rozhodně nejblíž.

[1] Prvních 7 Carmichaelových čísel ve skutečnosti objevil už český matematik Šimerka roku 1885, ale s jeho prací publikovanou v českém periodiku byl seznámen jen úzký okruh lidí.

Tabulka 2.1: $C(X)$ značí počet Carmichaelových čísel a $P(X)$ počet Fermatových pseudoprvočísel o základu 2, která jsou menší rovna X

X	$C(X)$	$P(X)$
10^4	7	22
10^5	16	78
10^6	43	245
10^7	105	750
10^8	255	2057
10^9	646	5597
10^{10}	1547	14884
10^{11}	3605	38975
10^{12}	8241	101629
10^{13}	19279	264239

Pro Carmichaelova čísla můžeme Fermatův test ztotožnit s testem hledajícím největšího společného dělitele — a ten žádnou záruku o spolehlivosti v padesáti procentech případů nenabízí. Carmichaelova čísla jsou ovšem poměrně řídká a tak je riziko chybného považování velkého čísla za prvočíslo poměrně malé. Pro ilustraci přikládáme tabulku dokumetující počet Carmichaelových čísel různých velikostí a také počet Fermatových pseudoprvočísel v bázi 2.

3 Solovayův-Strassenův test

Roku 1977 byl publikován Solovayův-Strassenův test prvočíselnosti [15]. V té době to byl velmi důležitý objev: tento test byl prvním zveřejněným probabilistickým algoritmem pro testování prvočíselnosti a poskytnul tak efektivní a velice spolehlivý způsob, jak získat velká prvočísla pro použití v kryptosystému RSA. Ačkoli je o pár let později zveřejněný Rabinův-Millerův prokazatelně lepší i rychlejší a od používání Solovayova-Strassenova algoritmu se brzy upustilo, pro svůj historický význam je algoritmus často zmiňován i v současnosti.

3.1 Matematické základy

Algoritmus pracuje s tzv. Legendreovým symbolem a jeho zobecněním, Jacobiho symbolem. Legendreův symbol byl zaveden již v roce 1798, Jacobiho symbol zhruba o 40 let později.

Definice (Legendreův symbol)**.** Nechť p je prvočíslo, $b \in \mathrm{Z}$. *Legendreův symbol* $(b \mid p)$ definujeme takto:

$$(b \mid p) = \begin{cases} 1 & \text{pokud } p \nmid b \text{ a existuje celé číslo } x \text{ tak, že } x^2 \equiv b \bmod p \\ 0 & \text{pokud } p \text{ dělí } b \\ -1 & \text{pokud } p \nmid b \text{ a neexistuje celé číslo } x \text{ tak, že } x^2 \equiv b \bmod p \end{cases}$$

Z hlediska terminologie můžeme poznamenat, že takové číslo b nesoudělné s prvočíslem p, které je druhou mocninou nějakého x modulo p, označujeme jako *kvadratické reziduum modulo p*.

Definice (Jacobiho symbol)**.** Nechť N je liché přirozené číslo s prvočíselným rozkladem $N = \prod_{i=1}^{k} p_i^{\alpha_i}$ a $M \in \mathrm{Z}$. *Jacobiho symbol* $(M \mid N) \in \{-1, 0, 1\}$ získáme jako součin odpovídajících Legendreových symbolů: $(M \mid N) = (b \mid p_1)^{\alpha_1} \ldots (b \mid p_k)^{\alpha_k}$.

Legendreův symbol je možné definovat také jinak: řekneme, že označuje hodnotu $b^{\frac{p-1}{2}} \bmod p$. Ekvivalence obou kvantit byla známa již před Legendrem a bývá označována jako Eulerovo kritérium.

Věta 6 (Eulerovo kritérium)**.** *Nechť p je prvočíslo, b celé číslo. Pak platí*

$$(b \mid p) \equiv b^{\frac{p-1}{2}} \bmod p$$

Důkaz. Případ $p \mid b$ je jasný: $(b \mid p) = 0$ z definice Legendreova symbolu, $b \bmod p = 0$ a umocnění číslem $\frac{p-1}{2}$ na tom nic nezmění.

Nechť tedy $p \nmid b$. Nechť r je primitivní kořen prvočísla p. Tedy existuje $i < p - 1$ tak, že $b = r^i \bmod p$. Rozlišujme dva případy: i může být liché nebo sudé.

A) Nechť $i = 2j$. Pak

$$b^{\frac{p-1}{2}} = r^{j(p-1)} = 1 \bmod p$$

A b má dva různé kořeny r^j a $r^{j+\frac{p-1}{2}}$: ${r^j}^2 = r^{2j} = r^i = b \bmod p$ a ${r^{j+\frac{p-1}{2}}}^2 = r^{2j+\frac{p-1}{2}} = r^{2j} r^{p-1} = b \bmod p$. Tohle platí právě pro polovinu možných b (stačí uvažovat pouze $b < p$). Jelikož každé má dva různé kořeny a my pracujeme modulo p, tak jsme už použili všechny možné kořeny. Z toho vyplývá, že pro liché exponenty musí platit, že taková b jsou kvadratickými nezbytky.

B) Nechť $i = 2j + 1$. Pak

$$b^{\frac{p-1}{2}} = r^{j(p-1)} r^{\frac{p-1}{2}} = -1 \bmod p$$

Poslední rovnost platí, protože $r^{j(p-1)} = 1 \bmod p$ a $r^{\frac{p-1}{2}}$ dává po umocnění na druhou jedničku, samo se ale nerovná jedné (protože r je primitivní kořen, jedinou zbývající možností je tedy $r^{\frac{p-1}{2}} = -1$. (Protože jediné číslo různé od jedničky, jehož umocněním na druhou modulo prvočíslo získáme jedničku, je -1). □

Právě z této vlastnosti prvočísel vychází celý Solovayův-Strassenův test. Aby byl takový test proveditelný, potřebujeme ověřit ještě několik věcí. Předně se musíme zamyslet nad platností Eulerova kritéria pro Jacobiho symboly — je nezbytné, aby pro každé testované složené číslo N existoval dostatek svědků, pro které vztah nebude platit. Přesné znění tohoto tvrzení a jeho důkaz bude obsahem dalšího oddílu. Zadruhé, je třeba najít algoritmus, který by Jacobiho symbol počítal v polynomiálním čase, tedy bez znalosti prvočíselného rozkladu čísla N. Vlastnostmi Legendreových a Jacobiho symbolů, které nám umožní takový algoritmus sestavit, se budeme zabývat právě teď.

3.1.1 Vlastnosti Jacobiho symbolu

Nejprve je třeba uvést a dokázat několik vlastností Legendrova symbolu. Později se ukáže, že všechny platí i pro Jacobiho symboly, a tuto platnost lze většinou odvodit jednoduchým zobecněním.

Lemma 2 (Vlastnosti Legendreova symbolu)**.** *Nechť n, m jsou lichá prvočísla. Potom platí:*
(a) $(ab \mid n) = (a \mid n)(b \mid n)$
(b) Když $a \equiv b \bmod n$*, pak* $(a \mid n) = (b \mid n)$
(c) $(2 \mid n) = (-1)^{\frac{n^2-1}{8}}$ *(tedy* $(2 \mid n)$ *má hodnotu* -1 *pouze pokud* $n \bmod 8$ *se rovná 3 nebo 5)*
Zákon kvadratické reciprocity
(d) $(m \mid n)(n \mid m) = (-1)^{\frac{m-1}{2}\frac{n-1}{2}}$ *(tedy* $(m \mid n)(n \mid m)$ *se rovná* -1 *pouze tehdy, když* $m \equiv n \equiv 3 \bmod 4$*)*

Vlastnosti (a) a (b) vyplývají pro Legendreův symbol přímo z definice a Eulerova kritéria — jsou to vlasnosti počítání s mocninami. Další dva body vyžadují důkladnější zamyšlení a vedou až k zajímavému, geometrickému důkazu zákona kvadratické reciprocity. Předtím potřebujeme vyslovit ještě dvě pomocná tvrzení.

Lemma 3 (Gaussovo lemma)**.** *Nechť p je prvočíslo, q je nesoudělné s p. Platí* $(q \mid p) = (-1)^m$*, kde m je počet čísel větších než* $\frac{p-1}{2}$ *v množině reziduí* $R = \left\{q, 2q, \ldots, \frac{p-1}{2}q\right\} \bmod p$.

Příklad Pro snazší pochopení obecného důkazu, který uvedeme vzápětí, začneme ilustrativním příkladem. Zvolme prvočísla $p = 31$ a $q = 19$. Prvky množiny R získáme jako $19 \bmod 31, 2 \cdot 19 \bmod 31, \ldots, 15 \cdot 19 \bmod 31$. Vychází $R = \{19, 7, 26, 14, 2, 21, 9, 28, 16, 4, 23, 11, 30, 18, 6\}$. Nyní vytvoříme množinu S tak, že prvky $a \in R$, které jsou větší než $\frac{p-1}{2} = 15$, nahradíme prvky $p - a$. Dostaneme $S = \{31 - 19, 7, 31 - 26, \ldots, 6\} =$
$\{12, 7, 5, 14, 2, 10, 9, 3, 15, 4, 8, 11, 1, 13, 6\}$. Vidíme, že množina S obsahuje právě všechna čísla mezi 1 a 15. To není náhoda, přesvědčit nás o tom může následující důkaz.

Důkaz. Pro prvky množiny R platí, že jsou navzájem různé a také žádné dva nemají součet p (kdyby $aq + bq = 0 \bmod p$, pak jelikož p je nesoudělné s q, tak p musí dělit $a + b$, což nejde: $a + b < p$). Definuji množinu S tak, že všechna $a \in R$, které jsou větší než $\frac{p-1}{2}$, nahradím hodnotou $p - a$. Množina S se tak od R bude lišit právě v m prvcích. Tvrdím, že množiny $\left\{1, 2, .., \frac{p-1}{2}\right\}$ a S jsou stejné. Všechny prvky S jsou evidentně menší rovny $\frac{p-1}{2}$ a žádné dva prvky S se nerovnají: kdyby ano, pak jeden z nich musel být už v R a druhý vzniknul jako $p - a$. Měli bychom tedy dva prvky R které by dávaly součet p, což není možné.
Můžeme psát $S = \left\{\pm q, \pm 2q, .., \pm\frac{p-1}{2}q\right\} \bmod p$, kde právě m prvků množiny má minusové znaménko. Nyní vynásobíme prvky obou množin.

$$(\frac{p-1}{2})! = (\frac{p-1}{2})! q^{\frac{p-1}{2}} (-1)^m \bmod p$$

Díky tomu, že $p \nmid (\frac{p-1}{2})!$, musí platit $q^{\frac{p-1}{2}} \bmod p = (-1)^m$, což dokazuje celé lemma. □

Gaussovo lemma nám umožní dokázat další vlastnost úvodního lemmatu.

Důkaz vlastnosti (c). Z Gaussova lemmatu máme existenci m takového, že $(2 \mid p) = (-1)^m$. Zkusíme toto m spočítat. Budeme tedy počítat počet čísel větších než $\frac{p-1}{2}$ v množině $I = \{2, 4, 6, \ldots, p-1\}$. Je zřejmé, že $\left|\left(\frac{p}{2}, p\right) \cap I\right|$ je stejná jako $\left|\left(\frac{p}{4}, \frac{p}{2}\right) \cap \mathrm{Z}\right|$. Můžeme psát $p = 8c + r$ a tento výraz dosadit do výrazu. Získáme $m = \left|\left(2c + \frac{r}{4}, 4c + \frac{r}{2}\right) \cap \mathrm{Z}\right|$. Pokud budeme počítat modulo 2, můžeme zanedbat sudé násobky c. Stačí nám tedy spočítat počet celých čísel v intervalu $\left(\frac{r}{4}, \frac{r}{2}\right)$ modulo 2. Ten je 0 pro $r = 1, 7$, což odpovídá případům, kdy $\frac{p^2-1}{8}$ je sudé číslo, a 1 pro $r = 3, 5$, tedy když $\frac{p^2-1}{8}$ je liché. Tím je tvrzení pro Legendreovy symboly dokázáno. □

Lemma 4 (Eisensteinovo lemma)**.** *Nechť p je prvočíslo, q je liché číslo nesoudělné s p. Nechť m má stejný význam jako v Gaussově lemmatu. Pak platí $m = \sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor \bmod 2$.*

Důkaz. Vezměme množinu S z Gaussova lemmatu. Vyjádřeme součet prvků S (označme ho N) dvěma různými způsoby. Jednoduché vyjádření je prostý součet prvků S jako $\sum_{i=1}^{\frac{p-1}{2}} i$. Druhou možností je popsat způsob, jakým jsme množinu S vytvořili. Pak $N = q \sum_{i=1}^{\frac{p-1}{2}} i - p \sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor + mp \bmod 2$. Zde první sčítanec odpovídá součtu prvků množiny R, druhý sčítanec vyjadřuje operaci modulo p a třetí záměnu celkem m prvků a za $p-a$. Jelikož uvažujeme modulo 2, opačné znaménko u těchto m prvků se neprojeví ($-1 = 1 \bmod 2$). Navíc p a q jsou lichá čísla, tedy jedničky modulo 2, což nám umožní výraz zjednodušit: $N = \sum_{i=1}^{\frac{p-1}{2}} i - \sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor + m \bmod 2$. Nyní položíme obě alternativní vyjádření do rovnosti.

$$\sum_{i=1}^{\frac{p-1}{2}} i = \sum_{i=1}^{\frac{p-1}{2}} i - \sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor + m \bmod 2$$

Snadnou úpravou výrazu získám konečný vztah $m = \sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor \bmod 2$. □

Nyní už přistoupíme k důkazu zákona kvadratické reciprocity. Ten říká, že součin dvou Jacobiho symbolů $(p \mid q)(q \mid p)$ se rovná -1 právě tehdy, pokud p i q dávají po dělení čtyřmi zbytek 3.

Lemma 5 (Zákon kvadratické reciprocity)**.** *Nechť p a q jsou prvočísla.*

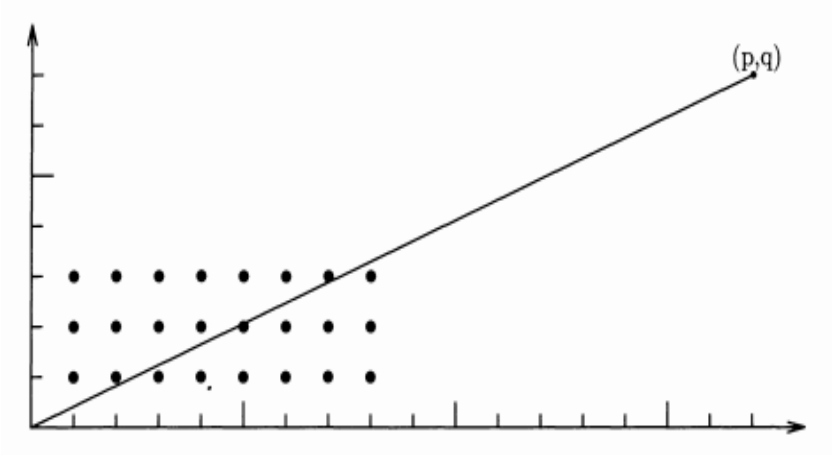
$$(p \mid q)(q \mid p) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$$

Důkaz. Ukážeme, že $\frac{p-1}{2}\frac{q-1}{2} = \sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor + \sum_{i=1}^{\frac{q-1}{2}} \lfloor \frac{pi}{q} \rfloor$. Tím bude celé lemma dokázáno: díky Eisensteinovu lemmatu

$$(-1)^{\sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor + \sum_{i=1}^{\frac{q-1}{2}} \lfloor \frac{pi}{q} \rfloor} = (-1)^{\sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor} (-1)^{\sum_{i=1}^{\frac{q-1}{2}} \lfloor \frac{pi}{q} \rfloor} = (q \mid p)(p \mid q)$$

Klíčovou rovnost je nejlepší chápat geometricky s pomocí obdélníka se stranami $\frac{q}{2}$ a $\frac{p}{2}$. Spočítáme, kolik leží uvnitř tohoto obdélníka bodů s oběma souřadnicemi celočíselnými. Hodnota x-ové souřadnice se musí pohybovat mezi 1 a $\frac{q-1}{2}$, y-ová souřadnice obdobně mezi 1 a $\frac{p-1}{2}$. Dohromady máme $\frac{p-1}{2}\frac{q-1}{2}$ bodů. Nyní obdélník rozdělme podle úhlopříčky na dva trojúhelníky. Úhlopříčka leží na přímce $y = \frac{p}{q}x$ a přímo na ní nenajdeme žádný celočíselný bod: p a q jsou nesoudělná. Součet počtu celočíselných bodů v horním a dolním trojúhelníku nám tak dá celkový počet takových bodů uvnitř obdélníka.

Pro ilustraci odvodíme počet takových bodů v dolním trojúhelníku, postup pro horní trojúhelník by byl zcela analogický. Budeme uvažovat přímky rovnoběžné s osou y, jež protínají osu x v bodě s x-ovou souřadnicí i, $1 \leq i \leq \frac{q-1}{2}$, i celé. Pro y-ovou souřadnici j bodů na jedné z těchto přímek, které leží uvnitř dolního trojúhelníka, musí platit $1 < j < \frac{p}{q}i$. Počet bodů na libovolné z úseček je tedy $\lfloor \frac{p}{q}i \rfloor$. Součet přes všechny úsečky nám dá počet bodů v dolním trojúhelníku jako $\sum_{i=1}^{\frac{q-1}{2}} \lfloor \frac{pi}{q} \rfloor$. Pro horní trojúhelník bychom dostali obdobně $\sum_{i=1}^{\frac{p-1}{2}} \lfloor \frac{qi}{p} \rfloor$. Počet celočíselných bodů v obdélníku je dán součtem těchto dvou výrazů: tím je věta o reciprocitě pro Legendreovy symboly dokázána. □



Obrázek 3.1: Body s celočíselnými souřadnicemi v obdélníku.[11]

Nyní budeme pracovat s Jacobiho symboly. Pro ujasnění zdůrazněme, že Jacobiho symbol už nemá žádný číselně teoretický význam: není svázán s kvadratickými zbytky ani s mocninami. Jeho hodnoty jsou stále v množině $\{-1, 0, 1\}$ a z výpočetního hlediska sdílí s Legendreovým symbolem mnohé vlastnosti. Ty nejdůležitější shrneme v následujícím lemmatu.

Lemma 6 (Vlastnosti Jacobiho symbolu)**.** *Nechť M a N jsou lichá vzájemně nesoudělná celá čísla.*
(a) $(M_1 M_2 \mid N) = (M_1 \mid N)(M_2 \mid N)$
(b) $(M \mid N) = (M \bmod N \mid N)$
(c) $(2 \mid N) = (-1)^{\frac{N^2-1}{8}}$
(d) (Zákon o kvadratické reciprocitě) $(M \mid N)(N \mid M) = (-1)^{\frac{M-1}{2}\frac{N-1}{2}}$

Ověření platnosti prvních dvou bodů nám nedá příliš práce.

Důkaz. Mějme prvočíselný rozklad čísla $n = \prod_{i=1}^{k} p_i^{\alpha_i}$. Podle definice Jacobiho symbolu můžeme rozepsat $(ab \mid n) = (ab \mid p_1)^{\alpha_1} \dots (ab \mid p_k)^{\alpha_k} = (a \mid p_1)^{\alpha_1}(b \mid p_1)^{\alpha_1} \dots (a \mid p_k)^{\alpha_k}(b \mid p_k)^{\alpha_k} = (a \mid n)(b \mid n)$. Druhá rovnost je tady vlastností Legendrových symbolů. Dokázali jsme tedy bod (a). Pro bod (b) si stačí uvědomit, že pokud $a \equiv b \bmod n$, pak pro každého prvočíselného dělitele čísla n také $a \equiv b \bmod p_i$.

I bod (c) dokážeme rozepsáním na součin Legendreových symbolů. Nechť $N = \prod p_i$, kde p_i jsou všechna prvočísla z prvočíselného rozkladu N (každé s příslušným počtem opakování). Podle definice Jacobiho symbolu můžeme rozepsat $(2 \mid N) = \prod (2 \mid p_i)$. Pro každý prvek součinu platí vztah známý pro Legendrovy symboly. Nechť P označuje počet p_i takových, že $p_i \bmod 8 \equiv \pm 3$. Pak $(2 \mid N) = (-1)^P$. Příslušný Jacobiho symbol se tedy rovná -1, pouze pokud P je liché. Zamyslíme se, co to znamená pro N. K tomu je dobré si uvědomit, že součin dvou čísel kongruentních s $\pm 1 \bmod 8$ je opět ± 1, stejně tak součin dvou čísel kongruentních s $\pm 3 \bmod 8$ je ± 1, a součin čísla kongruentního s ± 1 a čísla kongruentního s $\pm 3 \bmod 8$ je ± 3. Celý produkt $\prod_{p_i \equiv \pm 3 \bmod 8} p_i \bmod 8 = \pm 3$ právě tehdy, když P je liché. Vždy platí $\prod_{p_i \equiv \pm 1 \bmod 8} p_i \bmod 8 = \pm 1$. Součin těchto dvou produktů, tedy N, má tedy hodnotu ± 3 právě tehdy, když P je liché. Tím je věta dokázána: $(2 \mid N) = -1$ právě tehdy, když $N \equiv \pm 3 \bmod 8$, což je ekvivalentní tvrzení uvedenému v bodu (c) lemmatu.

Zbývá už jen zobecněný zákon kvadratické reciprocity. Nechť $M = \prod p_i$, $N = \prod q_j$, kde p_i a q_j jsou prvočísla (připouštíme opakování stejných prvočísel v jednom prvočíselném rozkladu). Rozepíšeme výraz $(M \mid N)(N \mid M) = \prod_{p_i, q_j} (p_i \mid q_j)(q_i \mid p_i)$. Použili jsme definici Jacobiho symbolu a bod (a). Spárováním stejných dvojic a aplikací věty o kvadratické reciprocitě získáváme

$(M \mid N)(N \mid M) = \prod_{p_i,q_j} (-1)^{\frac{p_i-1}{2}\frac{q_j-1}{2}}$. Zkusíme spočítat členy $(p_i \mid q_j)(q_j \mid p_i)$, jejichž hodnota je -1. To platí pouze v případě $p_i \equiv q_j \equiv 3 \bmod 4$. Nechť existuje K členů p_i takových, že $p_i \equiv 3 \bmod 4$ a L členů q_j takových, že $q_j \equiv 3 \bmod 4$. Pak dvojic, pro které je součin odpovídajících Jacobiho výrazů -1, je KL. Hodnota celého výrazu $(M \mid N)(N \mid M)$ je tudíž rovna jedné právě tehdy, když KL je liché.
Nyní se zamyslíme nad hodnotou $(-1)^{\frac{M-1}{2}\frac{N-1}{2}}$. Ta se rovná -1, právě tehdy když $M \equiv N \equiv 3 \bmod 4$. Pro výpočet $M \bmod 4$ využijeme prvočíselného rozkladu: $M = p_1 \dots p_m$ a $M \bmod 4 = 3^K 1^{(m-K)} = (-1)^K \bmod 4$. Díky tomu víme, že $M \equiv 3 \bmod 4$ právě tehdy když K je liché. Ze stejného důvodu platí $N \equiv 3 \bmod 4$ právě tehdy když L je liché. A zřejmě KL je liché pouze tehdy, když K je liché i L je liché.

Dohromady máme $(-1)^{\frac{M-1}{2}\frac{N-1}{2}} = -1$ právě tehdy když KL je liché právě tehdy když $(M \mid N)(N \mid M)$, a věta je dokázána. □

Pro Solovayův-Strassenův algoritmus je podstatné to, že je počitatelný v polynomiálním čase. Algoritmus na jeho výpočet využívá zákon kvadratické reciprocity a mírně tak připomíná Eukleidův algoritmus pro výpočet největšího společného dělitele.

Algoritmus 1 Výpočet Jacobiho symbolu: Jacobi(M, N)

$I \leftarrow 1$
while $M > 1$ **do**
  if $M > N$ **then**
    $M \leftarrow M \bmod N$
  end if
  Rozlož M na $M = 2^k L$, kde L je liché
  Aplikuj vztah pro $(2 \mid N)$:
  if $k \bmod 2 = 1$ **then**
    if $N \bmod 8 = 3$ nebo $N \bmod 8 = 5$ **then**
      $I \leftarrow -I$
    end if
  end if
  if $L = 1$ **then**
    return I
  end if
  Použij zákon kvadratické reciprocity:
  if $L \equiv N \equiv 3 \bmod 4$ **then**
    $I \leftarrow -I$
  end if
  Dál počítáme $(N \mid L)$:
  $M \leftarrow N$
  $N \leftarrow L$
end while
return $I \cdot M$

3.2 Popis algoritmu

Nejprve ukážeme konkrétní způsob použití pojmu, který jsme zmiňovali v úvodu povídání o pravděpodobnostních testech.

Definice (Svědek složenosti)**.** Mějme N liché složené celé číslo. Číslo $M < N$ nazvěme *svědkem složenosti čísla* N, pokud platí $(M \mid N) \neq M^{\frac{N-1}{2}} \bmod N$.

Minulá sekce představila novou vlastnost, kterou sdílejí všechna prvočísla. Narozdíl od Fermatova vztahu má tato vlastnost ještě jednu důležitou charakteristiku — pro každé složené číslo je vztah porušen v dostatečném počtu případů, existuje dostatečný počet svědků složenosti. Díky tomu je vhodná pro použití v pravděpodobnostním testu prvočíselnosti, jehož jednoduchý algoritmus nyní popíšeme.

Algoritmus 2 Solovayův-Strassenův test prvočíselnosti: Solovay-Strassen(M, N)

$N \leftarrow$ číslo, které chceme testovat na prvočíselnost
$M \leftarrow$ náhodné číslo mezi 2 a $N - 1$
if $gcd(M, N) > 1$ **then**
  return „N je složené číslo.“
else
Spočti $(M \mid N)$ a $M^{\frac{N-1}{2}} \bmod N$.
  if $(M \mid N) \neq M^{\frac{N-1}{2}} \bmod N$ **then**
    return „N je složené číslo.“
  end if
end if
return „N je pravděpodobně prvočíslo.“

Tento test můžeme opakovat pro různou volbu náhodného čísla M a tím významně snížit pravděpodobnost chyby, že nějaké složené číslo prohlásíme za prvočíslo. Při jednom opakování je pravděpodobnost této chyby maximálně 1/2, po K opakováních už pouze $\frac{1}{2^K}$.

Příklad fungování algoritmu

Pro lepší názornost předvedeme, jak by probíhal výpočet Solovayova-Strassenova algoritmu pro prvočíslo 3557.

Solovay-Strassen(3557,1000)
1. $1000^{1778} \equiv 1 \bmod 3557$
2. Výpočet Jacobiho symolu $Jacobi(1000, 3557)$:
Rozklad: $1000 = 2^3 125$
$3 \bmod 2 \equiv 1 \wedge 3557 \bmod 8 \equiv 5$, tedy $I = -1$. $3557 \bmod 4 \equiv 1$.
Dále počítáme $Jacobi(3557, 125) = Jacobi(57, 125)$. $57 \nmid 2$, $125 \bmod 4 \equiv 1$, pořád platí $I = -1$.
Dále počítáme $Jacobi(125, 57) = Jacobi(11, 57)$. $11 \nmid 2$, $57 \bmod 4 \equiv 1$, pořád platí $I = -1$.
Dále počítáme $Jacobi(57, 11) = Jacobi(2, 11)$. Rozklad: $2 = 2^1$, $11 \bmod 8 \equiv 3$, tedy $I = 1$ a výpočet je u konce.
Výsledky jsou oba rovny 1, 3557 je pravděpodobně prvočíslo.

3.3 Existence dostatečného počtu svědků

Algoritmus byl vysvětlen, nyní je potřeba ukázat, že opravdu funguje. To v případě pravděpodobnostího algoritmu znamená, že pro alespoň polovinu možných náhodných čísel použitých jako vstup algoritmu dá algoritmus správný výsledek. Díky tomu můžeme opakováním postupu snižovat pravděpodobnost chyby až k nule. Pravděpodobnostní povaha algoritmu pak nepředstavuje téměř žádné zvýšené riziko či omezení.

Důkaz bude rozdělen do dvou hlavních lemmat. Už víme, že pro prvočíslo vždy platí rovnost obou porovnávaných hodnot. Nejprve ukážeme, že pro čísla složená tomu tak není — vždy existuje alespoň jeden svědek složenosti. Začneme důkazem tvrzení o primitivních kořenenech modulo p^2.

Definice (Primitivní kořen modulo n)**.** Nechť n je libovolné celé číslo. Říkáme, že r je *primitivní kořen modulo* n, pokud pro všechna $m < \phi(n), m \mid \phi(n)$ platí, že $r^m \neq 1 \bmod n$.

Tvrzení 2. *Pro každé prvočíslo $p > 2$ existuje primitivní kořen modulo p^2.*

Důkaz. Využijeme znalost, že každé prvočíslo má primitivní kořen. Nechť r je primitivní kořen p. Ukážeme, že jedno z čísel $r, r + p$ je primitivní kořen modulo p^2.
Definujme číslo $s = r$, pokud $r^{p-1} \neq 1 \bmod p^2$, $s = r + p$ jinak. Jelikož $r + p = r \bmod p$, je $r + p$ také primitivní kořen p. Pomocí binomické věty spočtu pro druhý případ

$$(r+p)^{p-1} = r^{p-1} + (p-1)r^{p-2}p + \ldots + p^{p-1} = 1 + p(p-1)r^{p-2} \bmod p^2$$

Jelikož platí $p(p-1)r^{p-2} \neq 0 \bmod p^2$ (p je nesoudělné s $p-1$ i r^{p-2}), máme s takové, že $s^{p-1} \neq 1 \bmod p^2$. Označím stupeň s modulo p^2 jako j, tedy platí $s^j = 1 \bmod p^2$. Zároveň platí také $s^j = 1 \bmod p$, tedy $j = (p-1)k$. Z předchozího vyplývá, že $k \neq 1$. Stupeň s dělí velikost grupy, tedy $j \mid p(p-1)$. Dohromady máme $k \mid p$. Jelikož p je prvočíslo a $k \neq 1$, jedinou možností je $k = p$. Stupeň s je tedy $p(p-1)$, takže s je primitivní kořen modulo p^2. □

Lemma 7. *Nechť N je liché složené číslo. Pak existuje alespoň jeden svědek složenosti čísla N.*

Důkaz. Postupujeme sporem. Předpokláme, že existuje nějaké N bez svědků složenosti. Důkaz rozdělíme na dva případy.
(a) V prvočíselném rozkladu N se žádné číslo neopakuje, tedy $N = \prod_{i=1}^{k} p_i$. Mějme a t., že $(a \mid p_1) = -1$. Podle čínské zbytkové věty existuje b t., že $b =$

$a \bmod p_1, b = 1 \bmod p_i, i \in \{2, .., k\}$. Podle pravidel pro počítání s Jacobiho symboly platí:

$$(b \mid N) = \prod_{i=1}^{k} (b \mid p_i) = \prod_{i=1}^{k} (b \bmod p_i \mid p_i) = (a \mid p_1) \prod_{i=2}^{k} (1 \mid p_i) = -1.$$

Ale $b^{\frac{N-1}{2}} \bmod N \neq -1$. Kdyby ano, pak také $b^{\frac{N-1}{2}} \bmod p_2 = -1$, a to víme, že neplatí.
(b) Prvočíselný rozklad N obsahuje alespoň jednu druhou mocninu prvočísla, píšeme $N = p^2K$. Již jsme ukázali, že každé prvočíslo má primitivní kořen modulo p^2. Nechť r je primitivní kořen modulo p^2. Ukážeme, že $r^{N-1} \neq \pm 1 \bmod N$, tedy ani $r^{\frac{N-1}{2}} \neq \pm 1 \bmod N$ a rovnost s Jacobiho symbolem nemůže platit. Kdyby totiž $r^{N-1} \equiv 1 \bmod N$, pak z vlastností primitivních kořenů vyplývá, že $p(p-1) \mid N-1$. Evidentně p a $p-1$ jsou čísla nesoudělná, tedy také $p \mid N-1$. Zároveň víme $p \mid N$. Ale p nemůže dělit zároveň N a $N-1$, spor. □

Lemma 8. *Když N je liché složené číslo, pak pro alespoň polovinu $b \in \Phi(N)$ platí*

$$(b \mid N) \neq b^{\frac{N-1}{2}} \bmod N$$

Důkaz. Z minulého lemmatu máme existenci jednoho svědka složenosti b. Definuji množinu $A := \left\{a \in \Phi(N); (a \mid N) = a^{\frac{N-1}{2}} \bmod N\right\}$. Z ní odvodím množinu $bA = \{ba; a \in A\}$. Pro prvky množiny bA platí, že jsou navzájem různé a jsou také různé od všech prvků množiny A. To dokazuje celé lemma. Podrobněji: kdyby pro nějaké $a_1, a_2 \in A$ platilo $ba_1 = ba_2 \bmod N$, pak $b\,|a_1 - a_2| = 0 \bmod N$. Ale $b \in \Phi(N)$ a $|a_1 - a_2| < N$. Navíc pro všechna $a \in A : (ba \mid N) = (b \mid N)(a \mid N) \neq b^{\frac{N-1}{2}} a^{\frac{N-1}{2}} = ba^{\frac{N-1}{2}} \bmod N$. Množina A tedy nemůže obsahovat více než polovinu $b \in \Phi(N)$. □

4 Rabinův-Millerův test

S návrhem tohoto probabilistického testu jako první přišel M. Artjuhov kolem roku 1966. O něco později, ale nezávisle na něm objevil stejnou myšlenku J. L. Selfridge. Právě od něj pochází pojmenování pro složená čísla, která test při určité bázi neumí identifikovat, jakok „silná pseudoprvočísla“. Své jméno ovšem test získal až po autorech píšících o podobné idee o 10 let později než Artjuhov: deterministická verze tohoto testu, jejíž autorem je Miller, pochází z roku 1976 [10]. Tato verze testu prvočísel nicméně využívá předpoklad Rozšířené Riemannovy hypotézy (GRH). O čtyři roky později převedl Rabin test do probabilistické podoby, což umožnilo upustit od nedokázaného předpokladu GRH [12]. Rabinův-Millerův test je dodnes vnímán jako velmi efektivní způsob, jak testovat náhodně generovaná velká čísla na prvočíselnost.

Základní myšlenka testu je jednoduchým rozvinutím Malé Fermatovy věty. Tu použijeme k o něco propracovanější charakteristice prvočísel.

Věta 7. *Nechť p je liché prvočíslo. Pak $p-1$ je sudé a můžeme ho rozložit jako $p-1 = 2^s t$, kde t je liché. Potom pro každé $a \in \Phi(n)$ platí jedna z následujících dvou alternativ:*

$$a^t \equiv 1 \bmod p$$

nebo

$$\text{existuje nějaké } i,\ 0 \leq i \leq s-1 \text{ a platí } a^{2^i t} \equiv -1 \bmod p$$

Důkaz. Vezmeme p liché prvočíslo a číslo $a \in \Phi(n)$. Z Malé Fermatovy věty platí $a^{2^s t} \equiv 1 \bmod p$. Tento vztah můžeme po přepsání jedničky na levou stranu rozložit na $(a^{2^{s-1}t} - 1)(a^{2^{s-1}t} + 1) \equiv 0 \bmod p$. Jelikož p je prvočíslo, musí dělit jeden z uvedených součinitelů. Tedy buď platí, že $a^{2^{s-1}t} + 1 \equiv 0 \bmod p$, což znamená, že jsme nalezli vhodný exponent, pro který je splněna druhá podmínka věty, nebo $a^{2^{s-1}t} - 1 \equiv 0 \bmod p$. V tom případě můžeme opakovat rozklad na dva součinitele. Tímto postupem dosáhneme toho, že buď jednou nalezneme vhodný exponent i, že $a^{2^i t} \equiv -1 \bmod p$, nebo projdeme celý řetězec a vyjde nám, že $a^t \equiv 1 \bmod p$, což je první z podmínek uvedených ve větě. □

Objevili jsme tedy jednoduchou vlastnost, kterou sdílejí všechna prvočísla. Označme jí odpovídající relaci $R(n, a)$ a zapišme symbolicky celou definici:

$$R(n,a) \Leftrightarrow (a^t \equiv 1 \bmod n) \vee (\exists i \in \{0, s-1\} a^{2^i t} \equiv -1 \bmod n)$$

Abychom na této vlastnosti mohli založit probabilistický algoritmus na testování prvočíselnosti, zbývá se zamyslet nad její platností pro složená čísla. Totiž, že pro každé složené číslo n existuje dostatek čísel a takových, že $R(n, a)$ neplatí. Pro zjednodušení vyjadřování můžeme taková čísla opět nazývat svědci složenosti (čísla n). Naopak takové složené číslo n nesoudělné s číslem a, pro které platí $R(n, a)$, budeme označovat jako silné pseudoprvočíslo v bázi a.

Příklad Číslo 2 je svědek složenosti čísla $341 = 11 \cdot 31$. Platí totiž: $340 = 2^2 \cdot 85$, dále $2^{85} \equiv 32 \bmod 341$ a $2^{170} \equiv 1 \bmod 341$. Našli jsme tedy číslo, které není ± 1 a jehož umocněním na druhou získáme 1. To by při počítání modulo prvočíslo nebylo možné.

Číslo $91 = 7 \cdot 13$ je silné pseudoprvočíslo v bázích 9 a 10. Zdůvodnění: $90 = 2 \cdot 45, 10^{45} \equiv -1 \bmod 91, 9^{45} \equiv 1 \bmod 91$. Můžeme dokonce vyjmenovat všechny báze menší než 90, pro které je 91 silné pseudoprvočíslo. Jsou to: $9, 10, 12, 16, 17, 22, 29, 38, 53, 62, 69, 74, 75, 79, 81, 82$. V tomto konkrétním případě je poměr bází pseudoprvočísel (větší než 1 a menší než 90) a velikosti Eulerovy grupy čísel nesoudělných s 91 roven $16/72 \cong 0,22$. Pro číslo 341 existuje celkem 48 lživých bází, což dává poměr 0,16. Číslo 561 (které je nejmenším Carmichaelovým číslem) je silným pseudoprvočíslem pouze pro 8 bází.

4.1 Popis algoritmu

Předpokládejme, že těchto silných pseudoprvočísel není mnoho, Rabinův-Millerův algoritmus na testování prvočíselnosti pak můžeme popsat následujícím pseudokódem.

Algoritmus 3 Rabin-Miller(M, N)

$N \leftarrow$ číslo, které chceme testovat na prvočíselnost
$M \leftarrow$ náhodné číslo mezi 2 a $N - 1$
if $gcd(M, N) > 1$ **then**
  return „N je složené číslo.“
else
Vyjádři $N - 1$ jako $N - 1 = 2^s t$.
  if $M^t \equiv \pm 1 \bmod N$ **then**
    return „N je pravděpodobně prvočíslo.“
  else
    $i \leftarrow 1$
    while $M^{2^i t} \neq \pm 1 \bmod N$ AND $i < s - 1$ **do**
      $i \leftarrow i + 1$
    end while
    if $M^{2^i t} \equiv -1 \bmod N$ **then**
      return „N je pravděpodobně prvočíslo.“
    end if
    if $M^{2^i t} \equiv 1 \bmod N$ **then**
      return „N je složené číslo.“
    end if
  end if
  return „N je složené číslo“.

Příklad fungování algoritmu

Fungování Rabinova-Millerova algoritmu ilustrujeme na dvou jednoduchých případech.

Rabin-Miller(2,3473)
1. Rozklad: $3472 = 2^4 217$
2. Výpočet $2^t \bmod n$: $2^{217} \equiv 279 \bmod 3473$
3. Postupně počítáme $2^{2t}, \ldots, 2^{(s-1)t} \bmod n$: $2^{434} \equiv 1435 \bmod 3473$, $2^{868} \equiv 279 \bmod 3473$, $2^{1734} \equiv 236 \bmod 3473$. Ani jednou nevyšlo ± 1, test můžeme uzavřít a prohlásit, že 3473 je složené číslo.

Rabin-Miller(2,3557)
1. Rozklad: $3556 = 2^2 889$
2. Výpočet $2^t \bmod n$: $2^{889} \equiv 2614 \bmod 3557$
3. $2^{1778} \equiv -1 \bmod 3557$. V tuto chvíli můžeme prohlásit, že 3557 je pravděpodobně prvočíslo. Pravděpodobnost omylu snižujeme opakováním postupu

s jinou bází. Ukážeme ještě případ s bází 6.
2. Výpočet $6^t \bmod n$: $6^{889} \equiv -1 \bmod 3557$. Tento výpočet končí ještě rychleji, závěr je stejný: 3557 je pravděpodobně prvočíslo.

4.2 Existence dostatečného počtu svědků

K důkazu fungování uvedeného pravděpodobnostního algoritmu využijeme dvě lemmata, která nyní vysvětlíme. Pak už bude relativně snadé dokázat, že riziko, že po jednom opakování testu budeme složené číslo chybně považovat za prvočíslo, je menší jež 25 procent. Náš důkaz se nejvíc inspiruje důkazem uvedeným v [16]. Podobným způsobem je správnost Rabinova-Millerova testu ukázána i v [13].

Nejprve ukážeme vlastnost, kterou splňují všechna silná pseudoprvočísla.

Lemma 9. *Nechť n je liché přirozené číslo a 2^s nejvyšší mocnina dvojky taková, že $2^s \mid n-1$. Tedy $n-1 = 2^s t$, kde t je liché. Nechť dále m označuje nejvyšší exponent takový, že $2^m \mid p-1$ pro všechny prvočíselné dělitele p čísla n. Pokud n je silné pseudoprvočíslo v bázi a, platí buď $a^{2^{m-1}t} \equiv 1 \bmod n$, nebo $a^{2^{m-1}t} \equiv -1 \bmod n$.*

Důkaz. Díky předpokladu, že n je silné pseudoprvočíslo v bázi a, víme, že pro něj platí jedna z podmínek v definici $R(n, a)$. Pokud platí $a^t \equiv 1 \bmod n$, platí samozřejmě také $a^{2^{m-1}t} \equiv 1 \bmod n$, a lemma je tudíž splněno jednoduše. Dále můžeme předpokládat, že existuje i takové, že $a^{2^i t} \equiv -1 \bmod n$. Zvolme takové i a také fixujme prvočíselného dělitele čísla n (označme ho p). Platí $a^{2^i t} \equiv -1 \bmod p$ a zároveň $a^{2^{i+1}t} \equiv 1 \bmod p$. Nechť r je řád prvku a v grupě $\Phi(p)$. Připomeňme, že pokud r je řád prvku a, platí $a^r \equiv 1 \bmod p$ a navíc r je nejmenší takové číslo, pro které tohle platí. Pro řád prvku platí, že dělí všechna čísla s touto vlastností. Musí tedy platit $r \mid 2^{i+1}t$ a $r \nmid 2^i t$. Z toho odvodíme, že r obsahuje ve svém prvočíselém rozkladu 2 přesně v mocnině $i+1$: kdyby byla dvojka přítomna ve větší mocnině, neplatí $r \mid 2^{i+1}t$, kdyby naopak daná mocnina byla menší, muselo by platit i $r \mid 2^i t$. Z Malé Fermatovy věty máme $a^{p-1} \equiv 1 \bmod p$, a tedy pro r platí také $r \mid p-1$. Tato pozorování dohromady dávají $2^{i+1} \mid p-1$. Předchozí úvahy nejsou závislé na konkrétním p, $2^{i+1} \mid p-1$ tedy platí pro všechna p. Z definice m tedy platí nerovnost $m \geq i+1$. Tento vztah můžeme rozebrat postupně. Pokud platí rovnost, platí vztah $a^{2^{m-1}t} \equiv -1 \bmod n$, a věta je splněna. Pokud $m > i+1$, můžeme psát $m-1 = i+d$ pro d kladné číslo. Platí tedy $a^{2^{m-1}t} = a^{2^{i+d}t} = (a^{2^i t})^{2^d} \equiv 1 \bmod n$, což je přesně první podmínka z věty. Tvrzení je tímto dokázáno. □

S využitím minulého tvrzení nyní definujeme množinu, která obsahuje všechna silná pseudoprvočísla a její velikost jsme navíc schopni spočítat. K důkazu budeme potřebovat i fakt, že pro liché prvočíslo p a libovolný exponent α je grupa $\Phi(p^\alpha)$ cyklická. V minulé kapitole jsme předvedli, že takové tvrzení platí pro $\alpha = 2$, nyní tedy ukážeme obecnější verzi.

Věta 8. *Pro p liché prvočíslo a α přirozené číslo platí, že grupa $\Phi(p^\alpha)$ je cyklická.*

Důkaz. Víme, že tvrzení platí pro $alpha = 1$[1] a $\alpha = 2$. Větu budeme dokazovat indukcí: z předpokladu, že grupa $\Phi(p^\alpha)$ je cyklická, ukážeme, že i grupa $\Phi(p^{\alpha+1})$ je cyklická. Zvolme tedy $\alpha \geq 2$. Nechť g je generátor grupy $\Phi(p^\alpha)$. Zkusíme ukázat, že g je také generátor grupy $\Phi(p^{\alpha+1})$. Z definice g platí $g^{p^{\alpha-1}(p-1)} \equiv 1 \bmod p^\alpha$ a pro $i < p^{\alpha-1}(p-1)$ platí, že $g^i \not\equiv 1 \bmod p^\alpha$. Tedy i $g^{p^{\alpha-2}(p-1)} \not\equiv 1 \bmod p^\alpha$. Jelikož $p^{\alpha-2}(p-1) = \phi(p^{\alpha-1})$, Malá Fermatova věta nám dává $g^{p^{\alpha-2}(p-1)} \equiv 1 \bmod p^{\alpha-1}$. Můžeme tedy psát $g^{p^{\alpha-2}(p-1)} \equiv 1 + kp^{\alpha-1}$, kde $k \nmid p$.

Označme d řád p ve $\Phi(p^{\alpha+1})$. Z vlastností řádu prvku grupy musí platit $d \mid p^\alpha(p-1)$ a uvážíme-li zřejmou platnost vztahu $g^d \equiv 1 \bmod p^\alpha$, také $p^{\alpha-1}(p-1) \mid d$. Díky tomu máme jen dva možné kandidáty na d: buď $d = p^\alpha(p-1)$ nebo $d = p^{\alpha-1}(p-1)$. Druhou možnost bychom chtěli vyloučit: ukážeme, že $g^{p^{\alpha-1}(p-1)} \not\equiv 1 \bmod p^{\alpha+1}$. K tomu použijeme binomickou větu. Číslo $g^{p^{\alpha-1}(p-1)}$ napíšeme jako $(g^{p^{\alpha-2}(p-1)})^p = (1+kp^{\alpha-1})^p$. Tento výraz rozložíme:

$$(1 + kp^{\alpha-1})^p = 1 + pkp^{\alpha-1} + \binom{p}{2} k^2 p^{2(\alpha-1)} + \ldots + k^p p^{p(\alpha-1)}$$

. Všechny členy tohoto výrazu kromě prvních dvou jsou dělitelné $p^{\alpha+1}$: od čtvrtého členu dále je mocnina, ve které se vyskytuje p, alespoň $3(\alpha-1) \geq \alpha+1$ (uvažujeme $\alpha \geq 2$). Kombinační číslo ve třetím členu je dělitelné číslem p, příslušná mocnina u p je tedy po úpravě $2(\alpha-1)+1 \geq \alpha+1$. Platí tedy

$$(1 + kp^{\alpha-1})^p = 1 + pkp^{\alpha-1} \bmod p^{\alpha+1}.$$

Jelikož $p \nmid k$, platí také $(1+kp^{\alpha-1})^p \not\equiv 1 \bmod p^{\alpha+1}$. Tím je důkaz hotov. □

Lemma 10. *Nechť n je liché číslo, d je počet jeho prvočíselných dělitelů a m, t mají stejný význam jako v předchozím lemmatu. Navíc nechť $\overline{A}$ označuje následující množinu: $\overline{A} = \{0 \leq a < n; a^{2^{m-1}t} \equiv 1 \bmod n \vee a^{2^{m-1}t} \equiv -1 \bmod n\}$. Velikost množiny $\overline{A}$ je dána následujícím výrazem:*

$$\left|\overline{A}\right| = 2 \cdot 2^{(m-1)d} \prod_{p|n} gcd(t, p-1)$$

.

[1] Tohle tvrzení v naší práci bereme jako fakt.

Důkaz. Množinu $\overline{A}$ rozdělíme na dvě podmnožiny, které budeme uvažovat postupně. $\overline{A_1} = \{0 \leq a < n; a^{2^{m-1}t} \equiv 1 \bmod n$, $\overline{A_2} = \{0 \leq a < n; a^{2^{m-1}t} \equiv -1 \bmod n$, $\overline{A} = \overline{A_1} \cup \overline{A_2}$. Nechť $n = \prod p_i^{\alpha_i}$ je prvočíselný rozklad čísla n. Jako první určíme velikost množiny $\overline{A_1}$. To znamená, že budeme zkoumat počet řešení kongruence $a^{2^{m-1}t} \equiv 1 \bmod n$. S použitím Čínské zbytkové věty můžeme tuto rovnici rozložit na ekvivalentní soustavu rovnic ve tvaru $a^{2^{m-1}t} \equiv 1 \bmod p_i^{\alpha_i}$. Počet řešení soustavy je stejný jako počet řešení původní kongruence a můžeme ho získat jako součin počtu řešení jednotlivých kongruencí. Dále tedy budeme zvažovat jednu kongruenci tvaru $a^{2^{m-1}t} \equiv 1 \bmod p^{\alpha}$. Známý poznatek z teorie grup říká, že pro p prvočíslo a α kladné celé číslo je grupa $\Phi(p^{\alpha})$ cyklická. Každá cyklická grupa má generátor, označme tedy g generátor grupy $\Phi(p^{\alpha})$. Generátor je prvek maximálního řádu $\phi(p^{\alpha})$. Hodnota Eulerovy funkce $\phi(p^{\alpha})$ pro náš případ je $p^{\alpha-1}(p-1)$, všechna řešení uvažované kongruence lze tedy zapsat jako mocniny generátoru ve tvaru g^i, kde $i \leq p^{\alpha-1}(p-1)$.

Pro další úvahy budeme potřebovat největšího společného dělitele čísel $2^{m-1}t$ a $p^{\alpha-1}(p-1)$, označme ho D. Pokud g^i má být řešením, tedy splňovat $(g^i)^{2^{m-1}t} \equiv 1 \bmod p^{\alpha}$, $i2^{m-1}t$ musí být násobkem řádu generátoru $p^{\alpha-1}(p-1)$. S využitím D zapíšeme tento vztah jako $ixD = kyD$, přičemž $2^{m-1}t = xD$, $p^{\alpha-1}(p-1) = yD$ a čísla x a y jsou nesoudělná. Ekvivalentně tedy platí, že ix je násobkem y, a díky nesoudělnosti i je násobkem y. Jako řešení uvažované kongruence tedy můžeme dosadit všechny prvky tvaru g^i, kde i je násobkekm y: $y, 2y, \ldots, Dy = p^{\alpha-1}(p-1)$. Těchto řešení je přesně D. Vyjádření pro největšího společného dělitele čísel $2^{m-1}t$ a $p^{\alpha-1}(p-1)$ ještě můžeme trochu upravit: díky tomu, že $2^{m-1}t \mid n-1$ určitě víme, že čísla $2^{m-1}t$ a n jsou nesoudělná, tedy $2^{m-1}t$ je nesoudělené i se všemi prvočíselnými děliteli p čísla n. Z toho vyplývá $D = gcd(2^{m-1}t, p^{\alpha-1}(p-1)) = gcd(2^{m-1}t, (p-1))$. Dále z definice čísla m platí $2^{m-1} \mid (p-1)$ pro všechna p, což dovoluje konečnou úpravu vztahu na $D = 2^{m-1}gcd(t, p-1)$. Nyní už můžeme vyjádřit přímo počet prvků množiny $\overline{A_1}$ jako součin přes d prvočíselných dělitelů: $\left|\overline{A_1}\right| = 2^{(m-1)d} \prod_{p|n} gcd(t, p-1)$.

Tím jsme získali přesně polovinu z počtu prvků, které má obsahovat celá množina $\overline{A}$. Naší strategií nyní bude ukázat, že $\left|\overline{A_1}\right| = \left|\overline{A_2}\right|$, z čehož bude vyplývat platnost celého lemmatu. Pokud vztah definující množinu $\overline{A_2}$ rozdělíme na jednotlivé kongruence stejně, jako jsme postupovali v případě $\overline{A_1}$, dostaneme kongruence ve tvaru $a^{2^{m-1}t} \equiv -1 \bmod p^{\alpha}$. Ekvivalentně můžeme každou takovou kongruenci nahradit dvěma podobnými kongruencemi $a^{2^m t} \equiv 1 \bmod p^{\alpha}$ a $a^{2^{m-1}t} \not\equiv 1 \bmod p^{\alpha}$. Počet řešení první z nich bychom mohli odvodit stejným postupem, jaký jsme zde už použili, vyjde $2^m gcd(t, p-1)$. Od něj odečteme již známý počet řešení vztahu $a^{2^{m-1}t} \equiv 1 \bmod p^{\alpha}$ a

získáme tak počet řešení jedné z kongruencí jako

$$2^m gcd(t, p-1) - 2^{m-1} gcd(t, p-1) = 2^{m-1} gcd(t, p-1).$$

To je stejné jako v minulém případě. Násobením počtu řešení jednotlivých kongruencí bychom dostali počet řešení původního vztahu definujícího množinu $\overline{A_2}$. Z toho už je jasné, že $\left|\overline{A_1}\right| = \left|\overline{A_2}\right|$. □

Nyní nastal čas dokázat tvrzení, na kterém stojí celý Rabinův-Millerův algoritmus. Dokazovanou formulaci vyslovíme jako větu.

Věta 9. *Nechť n je liché složené číslo větší než 9 a A je množina všech bází menších než n, pro které je n silným pseudoprvočíslem. Potom pro velikost množiny A platí: $\frac{|A|}{\phi(n)} \leq 1/4$.*

Důkaz. V důkazu využijeme množinu $\overline{A}$, jejíž velikost jsme odvodili v minulém lemmatu. Díky lemmatu 9 víme, že $|A| \leq \left|\overline{A}\right|$, budeme tedy dokazovat vztah $\frac{\left|\overline{A}\right|}{\phi(n)} \leq 1/4$. Předpokládejme dále, že prvočíselný rozklad čísla n je $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_N^{\alpha_n}$. Podle vztahu v minulém lemmatu a známého vzorce pro výpočet Eulerovy funkce můžeme dokazované tvrzení rozepsat do tvaru

$$\frac{p_1^{\alpha_1 - 1} p_2^{\alpha_2 - 1} \dots p_N^{\alpha_N} (p_1 - 1)(p_2 - 1) \dots (p_N - 1)}{2 \cdot 2^{(m-1)d} \prod_{p|n} gcd(t, p-1)} \geq 4$$

Ještě provedeme menší úpravy na tvar

$$\frac{1}{2} \prod_{i=1}^{N} p_i^{\alpha_i - 1} \frac{(p_i - 1)}{2^{(m-1)} gcd(t, p_i - 1)} \geq 4$$

a podrobněji se podíváme na zlomek $\frac{(p_i-1)}{2^{(m-1)} gcd(t, p_i-1)}$. Jelikož $gcd(t, p_i - 1)$ je lichý dělitel $p_i - 1$ a hodnota výrazu $\frac{(p_i-1)}{2^{(m-1)}}$ je alespoň dva (protože $2^m \mid (p_i - 1)$), celý zlomek je vždy celé číslo ≥ 2.

Dále budeme postupovat rozborem případů, jak může vypadat prvočíselný rozklad čísla n. Nejjednodušší situace nastává, pokud n má tři a více prvočíselné dělitele. Výše analyzovaný zlomek se v celém výrazu vyskytuje nejméně třikrát, což po vydělení 2 dává minimum pro hodnotu výrazu 4, což nám úplně stačí.

Nechť tedy n má právě dva prvočíselné dělitele, $n = pq$. Uvažme nejdříve, že alespoň jedno z prvočísel, dejme tomu p, je v rozkladu n alespoň ve druhé mocnině. Pak $p^{\alpha-1} \geq 3$, což nám zaručuje hodnotu výrazu minimálně 6. Dále můžeme předpokládat, že mocnina obou prvočísel je 1. Zároveň

předpokládejme $p < q$. Nejprve uvažme, co by znamenala platnost vztahu $2^{m+1} \mid q-1$. Hodnota zlomku pro q by byla alespoň 4, což dohromady s minimem 2 pro p dává minimum celého výrazu 4, což zaručuje platnost tvrzení. Nechť tedy $2^{m+1} \nmid q-1$, píšeme $q-1 = 2^m q'$. Rozepišme $n-1 = pq-1 = p(q-1)+(p-1)$. Z toho je patrné, že $n-1 \bmod q-1 = p-1 \bmod q-1$. Jelikož $p < q$, $q-1$ nemůže být dělitelem $n-1$. Se sudou částí prvočíselného rozkladu problém být nemůže, neboť $2^m \mid 2^s$, zbývá tedy pouze možnost, že existuje lichý prvočíselný dělitel $q-1$, který dělí $q-1$ ve vyšší mocnině než dělí $n-1$. Díky téhle znalosti můžeme vznést omezení na největšího společného dělitele čísel t a $q-1$: $gcd(t, q-1) = gcd(t, q') \leq q'/3$. Pro zlomek $\frac{(q-1)}{2^{(m-1)}gcd(t,q-1)}$ díky tomu platí, že má hodnotu alespoň 6, a celý výraz je také minimálně 6.

Posledním zbývajícím případem je $n = p^\alpha$. V takovém případě víme, že α je alespoň dva, jinak by n bylo prvočíslo. Také víme, že 2^m je největší mocnina 2, která dělí $p-1$: $p-1 = 2^m p'$, kde p' je liché. Dále platí rovnost $n-1 = p^\alpha - 1 = (p-1)(p^{\alpha-1} + p^{\alpha-2} + \ldots + 1)$. Ta ukazuje, že $(p-1) \mid (n-1)$, neboli $(2^m p') \mid (2^s t)$. Aby tohle platilo, musí p' dělit t, $gcd(t, p-1) = p'$. Po dosazení do zlomku můžeme přesně určit jeho hodnotu: $\frac{(p-1)}{2^{(m-1)}gcd(t,p-1)} = 2$. Jelikož větu vyslovujeme pro $n > 9$, nejmenší prvočíslo, které se může vyskytovat ve druhé mocnině, je 5. Hodnota celého výrazu je tedy minimálně 5. Nyní už jsme prozkoumali všechny případy, důkaz je uzavřen.

4.3 Deterministická verze testu

V úvodu kapitoly jsme naznačili, že námi popisované verzi testu předcházela verze jiná, deterministická. Základní popis Millerova testu najdeme například v [4]. Bez toho, abychom se pouštěli do složité teorie okolo Rozšířené Riemannovy hypotézy, alespoň naznačíme myšlenku tohoto testu — ta je sama o sobě jednoduchá. Vychází z úvah o rozložení svědků složenosti mezi všemi bázemi. Kdyby totiž existovala nějaká horní mez taková, že nejmenší svědek složenosti pro dané číslo je vždy menší roven této bázi, měli bychom jasně dán maximální počet prvků, které je nutné otestovat, abychom prvočísla identifikovali s jistotou.

Pro většinu lichých složených čísel bude fungovat už nejmenší možný svědek, tedy 2. Dá se ovšem ukázat, že žádná univerzální horní mez neexistuje. Označme $W(n)$ nejmenšího svědka složenosti čísla n. Platí následující tvrzení:

Věta 10. *Existuje nekonečně mnoho lichých složených čísel n takových, že platí:*

$$W(n) \geq \ln^{\frac{1}{3 \ln \ln \ln(n)}}$$

I důkaz neexistence takové horní meze ještě nemusí znamenat úplné opuštění myšlenky hledání nejmenšího svědka. Stačilo by, kdyby se $W(n)$ podařilo omezit nějakou pomalu rostoucí funkcí. Pro složená čísla, jejichž prvočíselný rozklad obsahuje alespoň jedno prvočíslo nejméně ve druhé mocnině, platí toto:

Věta 11. *Nechť n je liché složené číslo dělitelné druhou mocninou nějakého prvočísla. Pak $W(n) < \ln^2(n)$.*

Pokud chceme vzít do úvahy všechna prvočísla, už se neobejdeme bez předpokladu Rozšířené Riemannovy hypotézy.

Věta 12. *Nechť n je liché složené číslo a platí Rozšířená Riemannova hypotéza. Pak $W(n) < 2\ln^2(n)$.*

□

5 Teorie probabilistických algoritmů

Předvedli jsme dva příklady pravděpodobnostních algoritmů na testování prvočíselnosti. Naše chápání pojmu „pravděpodobnostní algoritmus“ bylo až dosud spíše intuitivní: z popisu konkrétních zástupců bylo celkem jasné, proč se mluví o pravděpodobnosti. Nyní bychom se chtěli podívat na to, jak se takový pravděpodobnostní algoritmus rigorózně definuje. Představíme několik tříd, do kterých lze pravděpodobnostní algoritmy klasifikovat. V krátkosti naznačíme, jaké problémy s sebou může zavedení pravděpodobnosti do teorie algoritmů přinášet. Závěrem popíšeme některé možné aplikace. Celá kapitola by měla sloužit především jako přehled možných témat, kam může zkoumání pravděpodobnostních algoritmů pokračovat.

5.1 Intuitivní popis probabilistických algoritmů

Mohlo by se zdát, že spojení „pravděpodobnostní algoritmus“ nemůže dávat žádný smysl. Pod pojmem algoritmus si většinou představíme přesný postup, který nám krok po kroku říká, co máme provést. Správně použitý algoritmus vede k jednoznačnému výsledku. Naopak použití slova pravděpodobnost ihned navozuje pocit nejistoty ohledně výsledku.

Pravděpodobnostní algoritmus ale zdaleka není náhodný a toto spojení má při volbě správné definice jasný význam i odpovídající praktické použití. Dokonce by se dalo říci, že studium pravděpodobnostních algoritmů bylo přímo motivováno praxí, pro kterou klasické pojetí algoritmu a jeho složitosti není vždy optimální. Místo složitosti v nejhorším případě, s kterou zpravidla pracuje teorie výpočtové složitosti, nás v praxi zajímá spíše složitost průměrného případu — kterou ale bývá teoreticky obtížné určit. Přesto experimenty u mnohých *NP*-obtížných úloh prokazují, že typické instance těchto úloh jsou v praxi často řešitelné i pro poměrně velké vstupy. Jedním z

přístupů, jak se vypořádat se zdánlivě v reálném čase neřešitelnými úlohami, jsou probabilistické algoritmy. Každé vysvětlení pojmu pravděpodobnostní algoritmus musí obsahovat fakt, že pravděpodobnostní algoritmus poskytuje správný výsledek s určitou předem známou pravděpodobností, která je ohraničena dolní mezí tak, aby se při opakování algoritmu pravděpodobnost chyby vždy blížila nule. Náš výklad v této kapitole se bude rámcově držet kapitoly o pravděpodobnostních algoritmech v [9].

5.2 Formální zachycení probabilistického algoritmu

Jednu z možných formalizací pojmu algoritmus představuje použití Turingova stroje. Toto zařízení si obvykle představujeme jako spojení pásky, která bývá jedním směrem nekonečná, a čtecí hlavy, která čte znaky z pásky a podle nich a vnitřního stavu stroje se pohybuje jedním či druhým směrem, případně s přepsáním znaků na pásce. Chování čtecí hlavy při čtení symbolu určuje přechodová funkce, která je čistě deterministické povahy — při přečtení daného symbolu v jedné situaci je vždy přesně dáno, co má následovat.

Při zadání nějaké vstupní posloupnosti Turingova stroje je tedy průběh výpočtu přesně určen. Všechny možné Turingovy stroje pak vymezují jistou třídu vyčíslitelných funkcí. S tím souvisí známá Churchova teze, která říká, že pro každou efektivně vyčíslitelnou funkci (jako pojem odpovídající představě algoritmu) existuje Turingův stroj. Třída funkcí vymezených Turingovým strojem je prokazatelně stejná jako třída vyčíslitelných funkcí v jiných modelech „algoritmů“, např. teorii rekurzivních funkcí či přechodových diagramech. Churchova teze ale ze své povahy nejde dokázat — střetává se zde vágní představa pojmu algoritmus s přesně definovanou třídou funkcí vyčíslitelných konkrétním nástrojem.

Nyní ukážeme několik příkladů modifikace základní definice Turingova stroje: Turingův stroj s orákulem, jednoduchý nedeterministický Turingův stroj a pravděpodobnostní Turingův stroj.

Turingův stroj s orákulem

Jedním z možných modelů je *Turingův stroj s orákulem* [9]. Předpokládáme existenci vnějšího subjektu, orákula, který nám je schopen zodpovědět každou ano/ne otázku ohledně konkrétního rozhodovacícho problému. Na jeho odpovědi závisí další chování Turingova stroje. Orákulum se přitom chová jako černá skříňka — představujeme si, že je v jednom kroku schopné rozhodnout zadaný problém, a tento problém může být dokonce nerozhodnutelný.

Jedná se tedy o čistě teoretický model.

Formální model Turingova stroje s orákulem se může omezit na otázky, zda přirozené číslo dané okamžitým popisem na pásce v okamžiku položení otázky náleží do jisté množiny $A \subset \mathrm{N}$. Podle odpovědi na otázku pak Turingův stroj přejde do jednoho ze dvou specifikovaných stavů, nedochází přitom ani k pohybu hlavy, ani k zápisu žádného symbolu.

Pokud je množina A vyčíslitelná, shoduje se takto definovaná třída funkcí s funkcemi vyčíslitelnými pomocí obyčejného Turingova stroje — Turingův stroj by sám mohl zjistit, zda číslo na pásce náleží do A. I tak může postup využívající orákula znamenat urychlení výpočtu. V případě nerekurzivní množiny je třída funkcí vyčíslitelná s pomocí tohoto formalismu větší, orákulum tedy znamená opravdové rozšíření výpočetních možností Turingova stroje. Pomocí Turingových strojů s orákulem můžeme definovat systém složitostních tříd, které lze dále využít třeba pro zkoumání vztahu mezi P a NP.

Nedeterministický Turingův stroj

Nedeterminismus můžeme do Turingova stroje zavést takto: uvažujem TS jako množinu čtveřic ve tvaru $\langle q_i, S_j, X, q_l \rangle$, kde písmena po řadě odpovídají současnému vnitřnímu stavu, čtenému symbolu na pásce, akci a novému vnitřnímu stavu. Pro akci jsou na výběr tři základní možnosti: posun vlevo, posun vpravo, zápis symbolu na pásku. Nedeterministický stroj má navíc možnost čtvrtou: místo X doplníme vnitřní stav, a čtveřici chápeme tak, že stroj má na výběr, do kterého ze dvou nových vnitřních stavů přejde. Navíc povolíme, aby stroj obsahoval nekonzistentní čtveřice: tedy takové, které mají shodné první dvě pozice a liší se buď na třetí nebo čtvrté pozici.

Nabízí se dvojí interpretace takového stroje.

- Paralelní interpretace. V případě nalezení více použitelných čtveřic vytvoříme pro každou z nich vlastní větev, a algoritmus bude nadále běžet ve více větvích současně.
- Pravděpodobnostní interpretace. V okamžiku nalezení více použitelných čtveřic je vybrána jedna a pouze s tou výpočet pokračuje.

Pravděpodobnostní Turingův stroj

Pravděpodobnostní Turingův stroj by měl představovat variantu nedeterministického Turingova stroje. Můžeme ho definovat například takto: pravděpodobnostní Turingův stroj je Turingův stroj M se dvěma páskami, kde první páska funguje stejně jako páska u deterministického Turingova stroje

a druhá páska obsahuje náhodnou posloupnost 0 a 1. Přechod do dalšího stavu se určuje vždy na základě současného stavu a čtených symbolů na obou páskách. Jako akci je možné přesunout čtecí hlavu či přepsat symbol na první pásce, posloupnost 0 a 1 se ovšem nikdy nepřepisuje. S takto definovaným Turingovým strojem můžeme jednoduše popsat pravděpodobnostní složitostní třídy, které uvidíme v další části práce.

Definice pravděpodobnostního algoritmu

Pravděpodobnostní algoritmy můžeme modelovat i bez přímého použití Turingova stroje jako podmnožinu nedeterministických algoritmů. Abychom rozdíl mezi těmito dvěma pojmy ujasnili, ukážeme ještě jednu definici nedeterministického algoritmu a z ní odvozenou definici pro pravděpodobnostní algoritmus. Podrobnější popis i různé další varianty těchto pojmů najdeme v práci [8].

Definice (Nedeterministický algoritmus)**.** Nechť A, B, V jsou množiny, jejichž prvky je možné vzájemně jednoznačně očíslovat přirozenými čísly. Mějme dále funkci $f : A \to B$. Funkci $G_f : A \times V \to B$ nazveme nedeterministickým algoritmem pro výpočet funkce f, je-li G_f parciálně rekurzivní a pro každé $x \in D_f$ existuje $y \in V$ takové, že $G_f(x, y) = f(x)$.

Podmínka existence y podle definice je ale poměrně slabá. Pro praktické užití ji proto zesílíme a požadujeme jednu ze dvou následujících podmínek:
a) Pokud množina V je dostatečně malá, jsme schopni (nejlépe paralelně) spočítat hodnotu $G_f(x, y)$ pro všechna $y \in V$. Pokud navíc umíme mezi všemi hodnotami $G_f(x, y)$ rozpoznat správnou hodnotu $f(x)$, stává se výše definovaný nedeterministický algoritmus prakticky použitelným nástrojem.
b) Pokud pravděpodobnost, že pro náhodně zvolené y je hodnota $G_f(x, y)$ rovna hledané hodnotě $f(x)$, je vysoká. Jinak řečeno, požadujeme, aby v množině V převažovala ta y, která vedou ke správnému výsledku. Právě tahle větev vede k možné definici pravděpodobnostního algoritmu.

Pro situace, kdy má použití nedeterministického algoritmu smysl, by tedy měla platit alespoň jedna z podmínek a,b. Zároveň je třeba, aby složitost nedeterministického algoritmu, neboli (výpočetní) složitost funkce $G_f(x, y)$, byla menší než složitost funkce f. Právě úspora času je často hlavní důvod, proč upřednostňujeme nedeterministický algoritmus.

Samotná definice pravděpodobnostního algoritmu podle [8] vyžaduje některé technické pojmy, jejichž význam zde nebudeme vysvětlovat. Přesto je definice srozumitelná i na intuitiví úrovni a proto ji zde zmíníme.

Definice (Pravděpodobnostní algoritmus)**.** Mějme pravděpodobnostní prostor nad V $\langle V, B_0, \mu \rangle$, kde B_0 je $\sigma - algebra$ podmnožin množiny B a μ je

pravděpdobnostní míra na B_0. *Pravděpodobnostním algoritmem* pro výpočet funkce $f : D_f \to B$ s pravděpodobností chyby vzhledem k $\langle V, B_0, \mu \rangle$ stejnoměrně majorizovanou daným $\epsilon > 0$ nazveme totálně vyčíslitelnou funkci $G_f : A \times V \to B$ takovou, že pro každé $x \in D_f$ je $\mu(G_f(x, y))$ definována a $\mu(G_f(x, y)) \geq 1 - \epsilon$.

5.3 Třídy složitosti pro probabilistické algoritmy

Na základě probabilistického TM, který byl popsán výše, můžeme definovat několik tříd jazyků. Měnit se bude interpretace toho, co znamená, když Turingův stroj přijme nějaký vstup. V následujícím oddílu představíme několik tříd složitosti, do kterých můžeme klasifikovat pravděpodobnostní algoritmy. Bude vysvětlen i často používaný pojem Monte Carlo algoritmus. Naše prezentace čerpá zejména z knihy [11]. Historicky patří počátky zkoumání těchto tříd jazyků do stejné doby jako vznik popisovaných pravděpodobnostních algoritmů pro prvočíselnost — roku 1977 byly definovány J.Gillem v [6].

Budeme vycházet z následujících předpokladů: uvažujeme takové nedeterministické Turingovy stroje N, které mají v každém kroku na výběr právě ze dvou možností[1]. Délka výpočtu pro vstup x je vždy stejná a navíc polynomiální v x.

Definice (Polynomiální Monte Carlo Turingův stroj pro jazyk L)**.** Uvažujme nedeterministický Turingův stroj popsaný výše. Pro vstup délky n je délka jeho výpočtu polynomiální v n, pišme $p(n)$. O Monte Carlo TS mluvíme v případě, pokud navíc platí: pokud $x \in L$, pak alespoň polovina ze $2^{p(|x|)}$ výpočtů pro x dojde k výsledku, že $x \in L$. Pokud $x \notin L$, pak každý výpočet skončí s výsledkem „ne“.

Jinak řečeno, u Monte Carlo algoritmu nepřipouštíme jako výsledek žádná falešná pozitiva a pravděpodobnost chybné negativní odpovědi je ohraničena $1/2$.

Definice (Třída RP)**.** Třída RP (randomized polynomial time) obsahuje všechny jazyky, pro které existuje polynomiální Monte Carlo Turingův stroj.

Jednoduché pozorování říká, že místo pravděpodobnosti přijetí slova patřícího do L větší rovné $1/2$ bychom mohli uvažovat libovolné číslo mezi

[1] U reálných pravděpodobnostních algoritmů je většina kroků deterministických, to si v našem modelu můžeme představit jako hody mincí, kde bez ohledu na výsledek následuje vždy stejná akce.

0 a 1 a výsledná třída jazyků by se nezměnila. Konečný počet opakování algoritmu nám totiž vždy může zajistit pravděpodobnost chyby menší než 1/2, a dokonce i libovolně blízkou 0.

Logickým protějškem ke třídě RP je třída $coRP$, která nepřipouští žádné falešně negativní odpovědi a klade omezení na pravděpodobnost falešných pozitiv. Algoritmy, které patří do $RP \cap coRP$, pak mají velice příjemnou vlastnost, že bychom po určitém počtu opakování měli vždy dostat zaručeně správný výsledek. Buď totiž $x \in L$, a náš Monte Carlo algoritmus by to časem měl být schopen odhalit, nebo $x \notin L$, a v tom případě dříve či později dostaneme negativní odpověď od algoritmu z třídy $coRP$. Takové algoritmy nazýváme Las Vegas algoritmy.

Definice (Třída ZPP)**.** Třída ZPP (zero probability of error) obsahuje všechny jazyky, pro které existuje Las Vegas algoritmus.

Ačkoli z našich algoritmů to není patrné, úloha PRIMES patří do třídy ZPP. Námi představené algoritmy nikdy nevydají falešné tvzení, že x je složené číslo, a úloha prvočíselnosti podle nich patří do třídy $coRP$. Pro dokázání příslušnosti do třídy ZPP by tedy bylo třeba najít takový algoritmus, který se nikdy nemýlí při prohlášení, že x je prvočíslo. Takové algoritmy existují, ale využívají mnohem složitější teorii než námi popsané algoritmy [1].

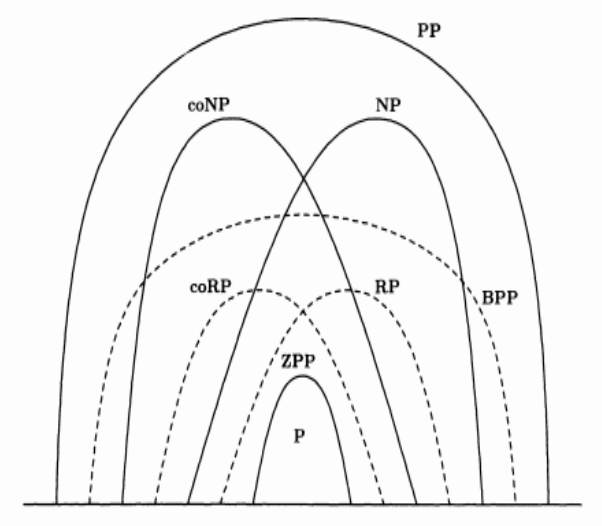
Mírně odlišné povahy je třída PP, ovšem i v jejím vymezení hraje roli pojem pravděpodobnosti. Narozdíl od předchozích tříd není třída PP vhodná pro praktické použití — rozdíl mezi $x \in L$ a $x \notin L$ může být jen v několika málo přijímajících výpočtech.

Definice (Třída PP)**.** Říkáme, že jazyk L patří do třídy PP, pokud existuje nedeterministický polynomiální Turingův stroj N takový, že pro každé x platí: x patří do L právě tehdy když více než polovina možných výpočtů stroje N skončí přijetím x.

Jako poslední uvedeme definici třídy BPP (bounded probability of error).

Definice (Třída BPP)**.** Jazyk L patří do třídy BPP, pokud pro něj existuje nedeterministický polynomiální Turingův stroj N s touto vlastností: pokud $x \in L$, pak alespoň 3/4 výpočtů N pro vstup x vedou k přijetí x. A pokud $x \notin L$, pak alespoň 3/4 výpočtů zamítnou x.

Definovali jsme čtyři nové třídy, do kterých je možno klasifikovat úlohy. Přirozenou otázkou nyní je, jaký je vztah mezi výše uvedenými třídami a jestli je můžeme nějak vnořit do hierarchie obsahující třídy P a NP. Snadno bychom zdůvodnili, že $ZPP \subseteq RP \subseteq BPP \subseteq PP$. Platí také $NP \subseteq PP$,



Obrázek 5.1: Třídy složitosti [11]. Tečkovaně jsou vyznačeny tzv. sémantické třídy složitosti: třídy, které nelze jednoduše standardizovat tak, aby o TM v odpovídající podobě bylo možné bez složitého uvažování rozhodnout, zda spadá do dané třídy.

$RP \subseteq NP$. Vztah mezi BPP a NP je nejasný. Zájemce o bližší zkoumání vztahů mezi těmito třídami můžeme odkázat například na text [17]. Autor zde předvádí alternativní definici všech tříd s pomocí kvantifikátorů $\forall, \exists$ a zobecněného kvantifikátoru $\exists^+$ s významem „jasné většiny“.

5.4 Problém náhodnosti

Další otázkou, nad kterou je vhodné se alespoň na chvilku zamyslet, je otázka náhodnosti. Důkazy fungování pravděpodobnostních algoritmů uvažují náhodná čísla. Jedním z aspektů náhodnosti je i to, že další číslo nijak nezávisí na posloupnosti čísel předchozích. V praxi ovšem žádný generátor náhodných čísel nemáme. Nabízí se sice využití jistých fyzikálních jevů, jenže to je těžko proveditelné. Skutečně se tedy spíše používají pseudonáhodné generátory. Takovému mechanismu dáme na začátku vstupní číslo (seed), a na jeho základě se podle určitého vzorce generují čísla další. Často se používá jednoduchá formule $x_{i+1} = (ax_i + b) \bmod c$, kde a, b, c jsou předem dané konstanty. Tímto způsobem vygenerovaná posloupnost čísel vykazuje při statistickém zkoumání vlastnosti velmi podobné náhodnosti, pravou náhodností ale nikdy nebude. Otázkou je, nakolik tento fakt ovlivní pravděpodobností algoritmy.

Filozofické zkoumání této otázky nabízí text [5]. Autor se zamýšlí nad

tím, zda pravděpodobnostní algoritmy mohou být použity nejen v praxi pro získání velkých prvočísel, ale také jako prostředek dokazování ve vědě či přímo matematice. Jako příklad, na kterém ilustruje celou problematiku, volí právě Rabinův algoritmus.

Aby obhájil spolehlivost pravděpodobnostních algoritmů, uvažuje autor několik možných cest. Ta první zkoumá empirické důkazy spolehlivosti těchto algoritmů. Fakt, že Rabinův algoritmus dává správný výsledek v případě, kdy jsme jeho správnost schopni ověřit alternativními metodami, ale není dostatečně pádným důkazem pro spolehlivost pravděpodobnostních algoritmů obecně. Přece jen jsme takto schopni ověřit jen malý zlomek všech možných případů.

Jinou možností je provést důkaz fungování pro konkrétní pravděpodobnostní algoritmus znovu, ale bez předpokladu pravých náhodných čísel. Tady se hodí zavést pojem *špatná posloupnost*, označující posloupnost čísel, kde všechny její prvky jsou lháři. Z předvedeného důkazu víme, že pravděpodobnost náhodného vygenerování takové posloupnosti je pro Rabinův algoritmus velmi malá. Co když je ale podíl takových posloupností vyšší mezi posloupnostmi získanými pomocí pseudonáhodného generátoru? Jakkoli se to zdá nepravděpodobné, není to úplně vyloučené — z celkem n^r možných posloupostí délky r generuje jednoduchý pseudonáhodný generátor popsaný výše pouze n sekvencí, každá z nich je určena prvním prvkem. Teoreticky jsou zde dvě možná úskalí. Mezi těmito n posloupnostmi by mohl být podíl špatných posloupností dost podstatný. Navíc, i malý počet špatných posloupností by mohl ohrozit výsledek algoritmu, pokud náš výběr vstupu pro generátor vede k preferování těchto špatných posloupností.

Vyvrátit tyto pochybnosti se pokoušel Bach [3]. Pro několik různých generátorů pseudonáhodných čísel dokazoval, že mezi jimi generovanými posloupnostmi je málo těch špatných. Pro náhodně zvolený první člen tak tento poznatek ukazuje, že o spolehlivosti Rabinova algoritmu nemusíme pochybovat. Potřeba pravých náhodných čísel tedy zůstává, byť v jasně menším rozsahu než dříve.

Nejlepším řešením by zřejmě bylo nahradit pseudonáhodný generátor něčím lepším. Zde se naše pozornost obrací k fyzikálním jevům, často ke kvantové mechanice. I tohle řešení s sebou nese řadu obtíží — generování čísel tímto způsobem může být pomalé a ani zde nemáme zaručenu dokonalou náhodnost. Autor dokonce vyslovuje tvrzení, že například hod mincí vůbec náhodným jevem není, neboť jeho výsledek je zcela určen počátečními podmínkami. To, že se nám výsledek jeví jako náhodný, je způsobeno tím, že nejsme schopni počáteční podmínky změřit a na jejich základě spočítat výsledek. Nepředvídatelnost ovšem není to samé co náhodnost.

I přes obsáhlý výčet problémů, se kterými se potýkáme při použití fy-

zických generátorů, není závěr textu nijak pesimistický. Není totiž žádný důvod obávat se, že mezi posloupnostmi získanými z takového generátoru bude zvýšený podíl špatných posloupností. Když budeme předpokládat, že tomu tak není, nedopouštíme se žádného pro vědu neobvyklého činu — předpoklady nepodložené empirickými fakty jsou často nezbytné. Nedůvěru k pravděpodobnostním algoritmům bude nejspíše dobré opustit.

5.4.1 Mírně náhodné generátory

Na závěr tohoto oddílu představíme jeden realistický model náhodnosti, který se zdá být dobře použitelný v praxi. Opět vycházíme z [11]. Popisovaným zdrojům náhodných čísel budeme říkat *mírně náhodné zdroje* (slightly random sources). Intuitivně jde o to, že jsme schopni přesně vymezit míru, s jakou se zdroj odchyluje od dokonalé náhodnosti.

Nechť δ je číslo splňující $0 < \delta \leq 1/2$ a p je funkce přiřazující posloupnostem 0 a 1 hodnotu z intervalu $[\delta, 1-\delta]$. Je dobré si p představovat jako komplexní funkci, o které nemáme žádné bližší informace. *δ-náhodný generátor* S_p je náhodná veličina s hodnotami v podobě nekonečných posloupností bitů, pro kterou platí: pravděpodobnost, že prvních n bitů má hodnoty $y_1, \ldots, y_n$ je dána jako

$$\prod_{i=1}^{n} (y_i p(y_1, \ldots, y_{i-1}) + (1 - y_i)(1 - p(y_1, \ldots, y_{i-1}))).$$

Pravděpodobnost toho, že i-tý bit nabývá hodnoty 1, je tedy přesně určena funkcí na předchozích hodnotách posloupnosti. Pro hodnotu $\delta = 1/2$ bychom měli dokonalý náhodný generátor, v praxi musíme očekávat hodnotu $\delta < 1/2$.

Pro zkoumání pravděpodobnostních algoritmů s pomocí δ-generátoru se nyní definují třídy $\delta - RP$ a $\delta - BPP$ odpovídající situaci, kdy uvažujeme všechny možné δ-náhodné zdroje, které všechny musí splňovat podmínky kladené na příslušnost do odpovídající třídy. Výsledkem této teorie je důležité tvrzení, které uvedeme bez důkazu.

Věta 13. *Nechť $\delta > 0$. Platí $\delta - RP = RP$ a $\delta - BPP = BPP$.*

5.5 Aplikace pravděpodobnostních metod

Zájem o pravděpodobnostní algoritmy vznikl právě v souvislosti s testováním prvočísel. V době vzniku popsaných pravděpodobnostních testů to byl jediný efektivní způsob, jak testovat prvočíselnost velkých čísel — aplikace pravděpodobnostního algoritmu umožnila efektivně řešit úlohu, která se zdála

být prakticky neřešitelná. I když dnes už máme k dispozici algoritmus AKS, který ukázal, že úloha prvočíselnosti patří do třídy P, pravděpodobností algoritmy stále zůstávají rychlejším způsobem, jak testovat prvočíselnost.

Jiné aplikace pravděpodobnostních algoritmů nejsou zdaleka tak známé. Pravděpodobnostní algoritmy nám přitom mohou pomoci i při řešení polynomiálních úloh. Jedním takovým příkladem je algoritmus na hledání vzoru v textu (pattern matching). Zadání bychom mohli formulovat takto: máme text délky X a úkolem je najít v něm první výskyt řetězce Y. Naivní přístup prochází text postupně v cyklu délky X. Vždy nejdříve porovná aktuální znak textu se znakem vzorku, v případě shody se posune v obou textech na další pozici a tenhle postup opakuje, dokud není ověřen souhlas celého vzorku s textem nebo dokud nedojde k neshodě. V tom případě se v cyklu posuneme a testování shody začíná znova od prvního znaku řetězce. Existuje množství propracovanějších deterministických algoritmů, my představíme probabilistickou verzi [7].

Použitá technika se nazývá *otisky* (fingerprinting). Idea spočívá v tom, že použijeme funkci, která řetězec délky Y nahradí něčím podstatně kratším (otiskem) a místo porovnávání řetězců pak porovnáváme tyto otisky. Jako funkci můžeme zvolit např. dělení modulo prvočíslo. Riziko spočívá v tom, že použitá funkce nebývá prostá, tzn. může přiřadit dvěma různým argumentům stejný otisk. To při porovnávání vede k tomu, že objevíme falešnou rovnost. Pokud je přesnost algoritmu nezbytná, lze to snadno řešit — nalezenou shodu otestujeme znak po znaku. Tyto dvě varianty algoritmu dobře ilustrují rozdíl mezi Monte Carlo a Las Vegas algoritmy — pokud přijmeme každou navrhovanou rovnost i s rizikem omylu, jedná se o Monte Carlo algoritmus. Pokud navíc rovnosti testujeme, algoritmus už patří do skupiny Las Vegas — je zde (malá) šance, že algoritmus bude potřebovat delší než polynomiální čas.

Užití pravděpodobnostních algoritmů pro praktické účely se zdá být dobře opodstatněné. Co kdybychom pravděpodobnostní algoritmus použili i k dokazování matematických vět? To znamená, že bychom nějakou větu prohlásili za dokázanou, i když bychom věděli, že je tu velmi malá pravděpodobnost omylu — třeba $1/2^{200}$. V téhle situaci už použití probabilistických metod tak bezproblémové nevypadá. Stejně tak bychom nejspíš o matematické větě nechtěli říkat, že je „skoro správná“.

Příbuzným tématem k dokazování jsou tzv. *protokoly s nulovou znalostí* (zero knowledge protocols). Uvažujme dva hráče, A a B. A potřebuje přesvědčit B, že má nějakou konkrétní informaci/zná řešení problému, ale bez toho, aby toto řešení prozradil. Situaci můžeme ilustrovat na problému obarvení grafu třemi barvami. A postupně vybírá úseky grafu, jejichž obarvení prozradí B. Mezi jednotlivými úseky ale může barvy měnit. S větším počtem

prozrazených úseků roste přesvědčení B o tom, že celý graf je možné obarvit třemi barvami, není však schopen takové obarvení jednoduše zrekonstruovat.

6 Závěr

Podali jsme podrobný popis dvou nejznámějších pravděpodobnostních algoritmů pro prvočíselnost a tím splnili základní cíl naší práce. Ukázalo se, že pochopení těchto algoritmů nevyžaduje žádné hluboké znalosti složité matematiky. Kdybychom ovšem chtěli zkoumat i původní, deterministickou verzi Rabinova-Millerova testu, dostali bychom se k Rozšířené Riemannově hypotéze a obtížnému zkoumání rozložení prvočísel mezi přirozenými čísly. Jiný směr, kam by zájemce o tuto problematiku mohl směřovat, je seznámení se s deterministickým testem AKS. Výklad tohoto algoritmu je možné nalézt už i v češtině, například v textu [14].

Mnoho prostoru ke zkoumání dosud panuje i v oblasti teorie pravděpodobnostních algoritmů. Podobně jako v případě problému $P = NP$?, i vztahy mezi pravděpodobnostními složitostními třídami jsou často nejasné. Těmto problémům se ale naše práce věnovat nechtěla. Naopak problematika, která by své místo v naší práci jistě měla, se týká reálných dat o používání jednotlivých algoritmů. Během našeho zkoumání se ukázalo zejména to, že takové informace — až na stručné zmínky o tom, že probabilistické algoritmy se i dnes běžně používají — ve vědeckých textech nenajdeme. Takové pátrání by možná bylo úspěšnější v knihovnách pro různé programovací jazyky či v kryptografických standardech.

Bibliografie

[1] L Adleman a M Huang. "Recognizing primes in random polynomial time". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM. 1987, s. 462–469.

[2] William R Alford, Andrew Granville a Carl Pomerance. "There are infinitely many Carmichael numbers". In: *The Annals of Mathematics* 139.3 (1994), s. 703–722.

[3] Eric Bach. "Realistic analysis of some randomized algorithms". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM. 1987, s. 453–461.

[4] Richard E Crandall a Carl Pomerance. *Prime numbers: a computational perspective*. Sv. 182. Springer, 2005.

[5] Don Fallis. "The reliability of randomized algorithms". In: *The British journal for the philosophy of science* 51.2 (2000), s. 255–271.

[6] John Gill. "Computational complexity of probabilistic Turing machines". In: *SIAM Journal on Computing* 6.4 (1977), s. 675–695.

[7] Richard M Karp a Michael O Rabin. "Efficient randomized pattern matching algorithms". In: *IBM Journal of Research and Development* 31.2 (1987), s. 249–260.

[8] Ivan Kramosil. "Parallel probabilistic searching and sortin algorithms". In: *Kybernetika* 26.Suppl (1990), s. 1–93.

[9] V Mařík, O Štěpánková a J Lažanský. *Umělá Inteligence (4)*. Academia, Praha, 2003.

[10] Gary L Miller. "Riemann's hypothesis and tests for primality". In: *Journal of computer and system sciences* 13.3 (1976), s. 300–317.

[11] Christos H Papadimitriou. *Computational complexity*. John Wiley a Sons Ltd., 2003.

[12] Michael O Rabin. "Probabilistic algorithm for testing primality". In: *Journal of number theory* 12.1 (1980), s. 128–138.

[13] René Schoof. "Four primality testing algorithms". In: *arXiv preprint arXiv:0801.3840* (2008).

[14] Jana Škarková. "Algoritmus AKS". Diplomová práce. Masarykova univerzita Brno, 2010.

[15] Robert Solovay a Volker Strassen. "A fast Monte-Carlo test for primality". In: *SIAM journal on Computing* 6.1 (1977), s. 84–85.

[16] Tomáš Váňa. "Silné pseudoprvočísla". Bakalářská práce. Univerzita Komenského, Bratislava, 2007.

[17] Stathis Zachos. "Probabilistic quantifiers and games". In: *Journal of Computer and System Sciences* 36.3 (1988), s. 433–451.