1.14.4 Třída $\mathcal{RP}$. Jazyk L patří do třídy $\mathcal{RP}$ právě tehdy, když existuje RTM M takový, že:

1. Jestliže $w \notin L$, stroj M se ve stavu q_f zastaví s pravděpodobností 0.
2. Jestliže $w \in L$, stroj M se ve stavu q_f zastaví s pravděpodobností, která je alespoň rovna $\frac{1}{2}$.
3. Existuje polynom $p(n)$ takový, že každý běh M (tj. pro jakýkoli obsah druhé pásky) trvá maximálně $p(n)$ kroků, kde n je délka vstupního slova.

Miller-Rabinův test prvočíselnosti je příklad algoritmu, který splňuje všechny tří podmínky (utvoříme-li k němu odpovídající RTM) a proto jazyk L, který se skladá ze všech složených čísel, patří do třídy $\mathcal{RP}$.

1.14.5 Turingův stroj typu Monte-Carlo. RTM splňující podmínky 1 a 2 z předchozí definice 1.14.4, se nazývá TM typu *Monte-Carlo*.

Uvědomte si, že RTM typu Monte-Carlo obecně nemusí pracovat v polynomiálním čase.

1.14.6 Tvrzení. Je dán jazyk $L \in \mathcal{RP}$, pak pro každou kladnou konstantu $0 < c < \frac{1}{2}$ je možné sestrojit RTM M (algoritmus) s polynomiální složitostí a takový, že:

1. Jestliže $w \notin L$, stroj M se úspěšně zastaví (tj. zastaví se ve stavu q_f) pravděpodobností 0.
2. Jestliže $w \in L$, stroj M se úspěšně zastaví (tj. zastaví se ve stavu q_f) s pravděpodobností aspoň $1 - c$.

1.14.7 Třída $\mathcal{ZPP}$. Jazyk L patří do třídy $\mathcal{ZPP}$ právě tehdy, když existuje RTM M takový, že:

1. Jestliže $w \notin L$, stroj M se úspěšně zastaví (tj. zastaví se ve stavu q_f) s pravděpodobností 0.
2. Jestliže $w \in L$, stroj M se úspěšně zastaví (tj. zastaví se ve stavu q_f) s pravděpodobností 1.
3. Střední hodnota počtu kroků M v jednom běhu je $p(n)$, kde $p(n)$ je polynom a n je délka vstupního slova.

To znamená: M neudělá chybu, ale nezaručujeme vždy polynomiální počet kroků při jednom běhu, pouze střední hodnota počtu kroků je polynomiální.

1.14.8 Turingův stroj typu Las-Vegas. RTM splňující podmínky z předchozí definice 1.14.7, se nazývá typu *Las-Vegas*.

1.14.9 Tvrzení. Jestliže jazyk L patří do třídy $\mathcal{ZPP}$, pak i jeho doplněk $\overline{L}$ patří do třídy $\mathcal{ZPP}$.

Stejný RTM M typu Las-Vegas slouží k přijetí jak jazyka L, tak i jeho doplňku $\overline{L}$; stačí koncové (přijímající) stavy RTM M prohlásit za nekoncové a ze všech nekoncových stavů M udělat koncové.

1.14.10 Poznámka. Pro jazyky ze třídy $\mathcal{RP}$ se tvrzení obdobné 1.14.9 neumí dokázat. To motivuje následující třídu jazyků.

1.14.11 Třída co-$\mathcal{RP}$. Jazyk L patří do třídy co-$\mathcal{RP}$ právě tehdy, když jeho doplněk $\overline{L}$ patří do třídy $\mathcal{RP}$.

1.14.12 Věta.

$$\mathcal{ZPP} = \mathcal{RP} \cap \text{ co-}\mathcal{RP}.$$

Nástin důkazu. Ukážeme nejprve $\mathcal{RP} \cap$ co-$\mathcal{RP} \subseteq \mathcal{ZPP}$.

Předpokládejme, že jazyk L leží v obou třídách $\mathcal{RP}$ i co-$\mathcal{RP}$. Existují proto dva RTM M_1 a M_2 typu Monte Carlo pracující v polynomiálním čase a takové, že

M_1 — přijímá jazyk L;

M_2 — přijímá jazyk $\overline{L}$.

Označme $p(n)$ ten větší z polynomů, které určují počet kroků M_1 a M_2. Setrojíme RTM M typu Las-Vegas, který přijímá L takto: Pro dané vstupní slovo w

1. M nechá pracovat M_1 po dobu $p(n)$ kroků. Jestliže M_1 přijme, M skončí a také přijme.
2. M nechá pracovat M_2 po dobu $p(n)$ kroků. Jestliže M_2 přijme, M skončí a nepřijme.
3. Jestliže M neskončí ani v kroku 1 ani v kroku 2, M pokračuje krokem 1.

Dá se dokázat, že RTM M je typu Las-Vegas.

Nyní ukážeme, že $\mathcal{ZPP} \subseteq \mathcal{RP} \cap$ co-$\mathcal{RP}$.

Předpokládejme, že jazyk L leží ve třídě $\mathcal{ZPP}$, existuje tedy RTM M_1 typu Las-Vegas, který přijímá jazyk L. Označme $p(n)$ polynom, který udává střední hodnotu počtu kroků RTM M_1 pro vstupní slovo délky n. Vytvoříme RTM M typu Monte Carlo pracující polynomiálně dlouho a přijímající jazyk L.

M nechá na vstupu w pracovat RTM M_1 po dobu $2p(n)$. Jestliže M_1 úspěšně skončí, M úspěšně skončí; ve všech ostatních případech RTM M skončí neúspěšně.

Dá se dokázat, že M splňuje všechny podmínky pro RTM typu Monte Carlo. Protože pracuje v čase $2p(n)$, jedná se o polynomiální RTM typu Monte Carlo. Proto je jazyk L ve třídě $\mathcal{RP}$.

Protože třída $\mathcal{ZPP}$ je uzavřena na doplňky, je každý jazyk ze třídy $\mathcal{ZPP}$ také ve třídě co-$\mathcal{RP}$.

1.14.13 Věta. Platí

$$\mathcal{P} \subseteq \mathcal{ZPP}, \quad \mathcal{RP} \subseteq \mathcal{NP}, \quad \text{co-}\mathcal{RP} \subseteq \text{ co-}\mathcal{NP}.$$

První inkluze je zřejmá, každý polynomiální Turingův stroj můžeme považovat za randomizovaný Turingův stroj typu Las-Vegas.

Druhá inkluze je složitější. Její důkaz spočívá v tom, že pro daný polynomiální RTM M typu Monte Carlo pracující v polynomiálním čase zkonstruujeme nedeterministický Turingův stroj, který přijímá stejný jazyk jako M.

Třetí inkluze jednoduše vyplývá z definic tříd co-$\mathcal{RP}$, co-$\mathcal{NP}$ a z druhé inkluze.

1.15 Nerozhodnutelnost

1.15.1 Rekursivní jazyky. Řekneme, že jazyk L je *rekursivní*, jestliže existuje Turingův stroj M, který rozhoduje jazyk L.

Připomeňme, že Turingův stroj M rozhoduje jazyk L znamená, že jej přijímá a na každém vstupu se zastaví (buď úspěwně nebo neúspěšně).

1.15.2 Rekursivně spočetné jazyky. Řekneme, že jazyk L je *rekursivně spočetný*, jestliže existuje Turingův stroj M, který tento jazyk přijímá.

Jinými slovy, M se pro každé slovo w, které patří do L, úspěšně zastaví a pro slovo w, které nepatří do L se buď zastaví neúspěšně nebo se nezastaví vůbec.

1.15.3 Poznámka. Jazykům, které nejsou rekursivní, také říkáme, že jsou *algoritmicky neřešitelné* nebo *nerozhodnutelné*. Obdobně mluvíme o úlohách, které jsou nerozhodnutelné nebo algoritmicky neřešitelné. První pojem se užívá častěji pro rozhodovací úlohy, druhý i pro úlohy konstrukční či optimalizační.

Každý rekursivní jazyk je též rekursivně spočetný. Ukážeme, že naopak to neplatí, tj. existují rekursivně spočetné jazyky, které nejsou rekursivní.

1.15.4 Tvrzení. Jestliže jazyk L je rekursivní, pak je rekursivní i jeho doplněk $\overline{L}$.

1.15.5 Tvrzení. Jestliže jazyk L i jeho doplněk $\overline{L}$ jsou oba rekursivně spočetné, pak L je rekursivní.

1.15.6 Tvrzení. Pro jazyk L může nastat jedna z následujících možností:

1. L i $\overline{L}$ jsou oba rekursivní.
2. Jeden z L a $\overline{L}$ je rekursivně spočetný a druhý není rekursivně spočetný.
3. L i $\overline{L}$ nejsou rekursivně spočetné.

1.15.7 Kód Turingova stroje. Každý Turingův stroj M lze zakódovat jako binární slovo. Mějme Turingův stroj M s množinou stavů $Q = \{q_1, q_2, \ldots, q_n\}$, množinou vstupních symbolů $\Sigma = \{0, 1\}$, množinou páskových symbolů $\Gamma = \{X_1, X_2, \ldots, X_m\}$, kde $X_1 = 0$, $X_2 = 1$ a $X_3 = B$. Dále počáteční stav je stav q_1, koncový stav je q_2. Označme D_1 pohyb hlavy doprava a D_2 pohyb hlavy doleva. (Tj. $D_1 = R$ a $D_2 = L$.)

Jeden přechod stroje M

$$\delta(q_i, X_j) = (q_k, X_l, D_r)$$

zakódujeme slovem

$$w = 0^i 10^j 10^k 10^l 10^r.$$

které nazýváme *Kód Turingova stroje* M, značíme jej $\langle M \rangle$, je

$$\langle M \rangle = 111\, w_1\, 11\, w_2\, 11 \ldots 11\, w_p\, 111,$$

Kde $w_1, \ldots, w_p$ jsou slova odpovídající všem přechodům stroje M.

1.15.8 Binární slova můžeme uspořádat do posloupnosti a tudíž je očíslovat. K binárnímu slovu w utvoříme $1w$ a toto chápeme jako binární zápis přirozeného čísla.

Tedy např. ϵ je první slovo, 0 je druhé slovo, 1 je třetí slovo, atd, 100110 je $1100110 = 64 + 32 + 4 + 2 = 102$, tj. 100110 je 102-hé slovo. V dalším textu o binárním slovu na místě i mluvíme jako o slovu w_i. Tedy $w_1 = \epsilon$, $w_{102} = 100110$.

Jedná se vlastně o uspořádání slov nejprve podle délky a mezi slovy stejné délky o lexikografické uspořádání.

1.15.9 Diagonální jazyk L_d. Nejprve uděláme následující úmluvu. Jestliže binární slovo w nemá tvar z 1.15.7, považujeme ho za kód Turingova stroje M, který nepřijímá žádné slovo. Tj. $L(M) = \emptyset$.

Jazyk L_d se skládá ze všech binárních slov w takových, že Turingův stroj s kódem w nepřijímá slovo w. (Tedy L_d obsahuje i všechna slova w, která neodpovídají kódům nějakého Turingova stroje, ovšem obsahuje i další binární slova.)

1.15.10 Věta. Neexistuje Turingův stroj, který by přijímal jazyk L_d. Jinými slovy, $L_d \neq L(M)$ pro každý Turingův stroj M.

Nástin důkazu. Postupujeme sporem. Kdyby existoval Turingův stroj M takový, že $L_d = L(M)$, pak by tento Turingův stroj měl kód roven nějakému binárnímu slovu, tj. $\langle M \rangle = w_i$ pro nějaké i.

Na otázku, zda toto slovo w_i patří nebo nepatří do jazyka L_d, nemůžeme dát odpověď, která by nevedla ke sporu.

Kdyby $w_i \in L_d$, pak w_i splňuje podmínku: Turingův stroj s kódem w_i nepřijímá slovo w_i. Ale $L_d = L(M)$ kde $w_i = \langle M \rangle$ — spor.

Kdyby $w_i \notin L_d$, pak Turingův stroj s kódem w_i nepřijímá slovo w_i. Ale to je podmínka pro to, aby slovo w_i patřilo do L_d — spor.

Proto neexistuje Turingův stroj, který by přijímal jazyk L_d.