1.11 Třída co-$\mathcal{NP}$

1.11.1 Pozorování. Je-li jazyk L ve třídě $\mathcal{P}$, pak i jeho doplněk $\overline{L}$ patří do třídy $\mathcal{P}$. Obdobné tvrzení se pro jazyky třídy $\mathcal{NP}$ neumí dokázat.

1.11.2 Definice. Jazyk L patří do třídy co-$\mathcal{NP}$, jestliže jeho doplněk patří do třídy $\mathcal{NP}$.

1.11.3 Příklady.

- Jazyk $USAT$, který je doplňkem jazyka SAT splnitelných booleovských formulí, leží ve třídě $co - \mathcal{NP}$. (Jazyk $USAT$ se skládá ze všech nesplnitelných booleovských formulí a ze všech slov, které neodpovídají booleovské formuli.)
- Jazyk $TAUT$, který se skládá ze všech slov odpovídajících tautologii výrokové logiky, patří do třídy $co - \mathcal{NP}$.

1.11.4 Otázka, zda co-$\mathcal{NP} = \mathcal{NP}$, je otevřená.

1.11.5 Tvrzení. co-$\mathcal{NP} = \mathcal{NP}$ právě tehdy, když existuje $\mathcal{NP}$ úplný jazyk, jehož doplněk je ve třídě $\mathcal{NP}$.

1.12 Třídy $\mathcal{P}SPACE$ a $\mathcal{NP}SPACE$

1.12.1 Je dán Turingův stroj M (determininstický nebo nedeterministický). Připomeňme, že M pracuje s paměťovou složitostí $p(n)$ právě tehdy, když pro každé slovo délky n nepoužije paměťovou buňku větší než $p(n)$.

1.12.2 Třída $\mathcal{P}SPACE$. Jazyk L patří do třídy $\mathcal{P}SPACE$ právě tehdy, když existuje deterministický Turingův stroj M, který přijímá jazyk L a pracuje s polynomiální paměťovou složitostí.

1.12.3 Tvrzení. Platí
$$\mathcal{P} \subseteq \mathcal{P}SPACE.$$

1.12.4 Třída $\mathcal{NP}SPACE$. Jazyk L patří do třídy $\mathcal{NP}SPACE$ právě tehdy, když existuje nedeterministický Turingův stroj M, který přijímá jazyk L a pracuje s polynomiální paměťovou složitostí.

1.12.5 Tvrzení. Platí
$$\mathcal{NP} \subseteq \mathcal{NP}SPACE.$$

1.12.6 Věta. Je dán Turingův stroj M (deterministický nebo nedeterministický), který přijímá jazyk L s paměťovou složitostí $p(n)$ (kde p je nějaký polynom). Pak existuje konstanta c taková, že M přijme slovo w délky n po nejvýše $c^{p(n)+1}$ krocích.

1.12.7 Myšlenka důkazu věty 1.12.6. Konstantu c volíme tak, abychom měli zajištěno, že Turingův stroj M má při práci se vstupem délky n méně než $c^{p(n)+1}$ různých konfigurací. Zajímají nás totiž pouze takové výpočty, ve kterých se konfigurace neopakují. Označme t počet páskových symbolů Turingova stroje M a označme s počet stavů M. Pak M má $p(n)\, s\, t^{p(n)}$ různých konfigurací.

Položme $c = t + s$. Z binomické věty vyplývá, že

$$c^{p(n)+1} = (t+s)^{p(n)+1} = t^{p(n)+1} + p(n)\, t^{p(n)}\, s + \dots.$$

Odtud $c^{p(n)+1} \geq p(n)\, t^{p(n)}\, s$.

1.12.8 Věta. Je-li jazyk L ve třídě $\mathcal{PSPACE}$ ($\mathcal{NPSPACE}$)), pak L je rozhodován deterministickým (nedeterministickým) Turingovým strojem M s polynomiální paměťovou složitostí, který se vždy zastaví po nejvýše $c^{q(n)}$ krocích, kde $q(n)$ je vhodný polynom a c konstanta.

1.12.9 Myšlenka důkazy věty 1.12.8. Předpokládejme, že $L \in \mathcal{PSPACE}$. Pak existuje Turingův stroj M_1, který přijímá jazyk L s paměťovou složitostí $p(n)$ ($p(n)$ je vhodný polynom). Víme (z věty 1.12.6), že existuje konstanta c taková, že Turingův stroj M_1 potřebuje nejvýše $c^{p(n)+1}$ kroků.

Vytvoříme Turingův stroj M_2, který bude mít dvě pásky: první páska bude simulovat M_1, druhá bude počítat kroky na první pásce. Jestliže počet kroků překročí $c^{p(n)+1}$, Turingův stroj M_2 se neúspěsně zastaví.

Hledaný Turingův stroj M je Turingův stroj s jednou páskou, který simuluje Turingův stroj M_2. Turingův stroj M pracuje v s časovou složitostí $\mathcal{O}(c^{2p(n)})$, tedy v maximálně $d\, c^{2p(n)}$ krocích. Nyní stačí položit $q(n) = 2p(n) + \log_c d$ nebo jakýkoli polynom větší.

1.12.10 Savitchova věta. Platí

$$\mathcal{PSPACE} = \mathcal{NPSPACE}.$$

1.12.11 Nástin myšlenky důkazu Savitchovy věty. Zřejmě $\mathcal{PSPACE} \subseteq \mathcal{NPSPACE}$. Důkaz opačné inkluze $\mathcal{NPSPACE} \subseteq \mathcal{PSPACE}$ spočívá v tom, že jsme schopni nedeterministický Turingův stroj pracující s paměťovou složitostí $p(n)$ simulovat deterministickým Turingovým strojem, který pracuje s paměťovou složitostí $\mathcal{O}([p(n)]^2)$.

Je dán nedeterministický Turingův stroj M, který přijímá jazyk L s polynomiální paměťovou složitostí $p(n)$. Konstrukce deterministického Turingova stroje přijímajícího stejný jazyk jako M s polynomiální paměťovou složitostí je založena na rekursivní proceduře $\mathit{dostup}(I, J, m)$, kde I a J jsou konfigurace a m je číslo. Výstup procedury $\mathit{dostup}(I, J, m)$ je buď 1, jestliže Turingův stroj se z konfigurace I do konfigurace J dostane v nejvýše m krocích, 0 v opačném případě. Procedura $\mathit{dostup}(I, J, m)$ pro každou konfiguraci K rekursivně zavolá procedury $\mathit{dostup}(I, K, m/2)$ a $\mathit{dostup}(K, J, m/2)$.

Pro vstup w voláme proceduru $\mathit{dostup}(I_0, J, m)$, kde I_0 je počáteční konfigurace M, J je přijímající konfigurace M a $m = c^{p(n)+1}$ (c je konstanta z 1.12.6). Dá se dokázat, že pro vykonání procedury $\mathit{dostup}(I, J, m)$ deterministickým Turingovým strojem stačí paměťová složitost $\mathcal{O}([h(n)]^2)$, kde $h(n)$ je vhodný polynom. (Uvědomte si, že nám nezáleží na tom, jak dlouho deterministický Turingův stroj pracuje, zajímáme se pouze o paměťové nároky.)

1.12.12 Důsledek. Platí

$$\mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{P}SPACE.$$