1.9 Třída $\mathcal{NPC}$

1.9.1 Redukce a polynomiální redukce úloh. Jsou dány dvě rozhodovací úlohy $\mathcal{U}$ a $\mathcal{V}$. Řekneme, že úloha $\mathcal{U}$ se *redukuje* na úlohu $\mathcal{V}$, jestliže existuje algoritmus (program pro RAM, Turingův stroj) M, který pro každou instanci I úlohy $\mathcal{U}$ zkonstruuje instanci I' úlohy $\mathcal{V}$ a to tak, že

$$I \text{ je ANO-instance } \mathcal{U} \text{ iff } I' \text{ je ANO-instance } \mathcal{V}.$$

Fakt, že úloha $\mathcal{U}$ se redukuje na úlohy $\mathcal{V}$ značíme

$$\mathcal{U} \lhd \mathcal{V}.$$

Jestliže navíc, algoritmus M pracuje v polynomiálním čase, říkáme, že $\mathcal{U}$ se *polynomiálně* redukuje na $\mathcal{V}$ a značíme

$$\mathcal{U} \lhd_p \mathcal{V}.$$

Fakt, že se úloha $\mathcal{U}$ redukuje na úlohu $\mathcal{V}$ zhruba řečeno znamená, že $\mathcal{U}$ není obtížnější než $\mathcal{V}$.

1.9.2 Tvrzení. Jsou dány tři rozhodovací úlohy $\mathcal{U}$, $\mathcal{V}$ a $\mathcal{W}$. Jestliže platí

$$\mathcal{U} \lhd_p \mathcal{V} \ \text{ a } \ \mathcal{V} \lhd_p \mathcal{W}, \quad \text{pak} \quad \mathcal{U} \lhd_p \mathcal{W}.$$

1.9.3 $\mathcal{NP}$ úplné úlohy. Řekneme, že rozhodovací úloha $\mathcal{U}$ je $\mathcal{NP}$ *úplná*, jestliže

1. $\mathcal{U}$ je ve třídě $\mathcal{NP}$;
2. každá $\mathcal{NP}$ úloha se polynomiálně redukuje na $\mathcal{U}$.

Třída všech $\mathcal{NP}$ úplných úloh se značí $\mathcal{NPC}$.

Zhruba řečeno, $\mathcal{NP}$ úplné úlohy jsou ty „nejtěžší“ mezi všemi $\mathcal{NP}$ úlohami.

1.9.4 Tvrzení. Jsou dány dvě $\mathcal{NP}$ úlohy $\mathcal{U}$ a $\mathcal{V}$, pro které platí $\mathcal{U} \lhd_p \mathcal{V}$. Pak

1. jestliže $\mathcal{V}$ je ve třídě $\mathcal{P}$, pak také $\mathcal{U}$ je ve třídě $\mathcal{P}$;
2. jestliže $\mathcal{U}$ je $\mathcal{NP}$ úplná úloha, pak také $\mathcal{V}$ je $\mathcal{NP}$ úplná úloha.

1.9.5 Tvrzení. Kdyby by některá $\mathcal{NP}$ úplná úloha patřila do třídy $\mathcal{P}$ (tj. byla by polynomiálně řešitelná), pak $\mathcal{P} = \mathcal{NP}$. Jinými slovy, každá $\mathcal{NP}$ úloha by byla polynomiálně řešitelná.

1.9.6 $\mathcal{NP}$ obtížné úlohy. Jestliže o některé úloze $\mathcal{U}$ pouze víme, že se na ní polynomiálně redukuje některá $\mathcal{NP}$ úplná úloha, pak říkáme, že $\mathcal{U}$ je $\mathcal{NP}$ těžká, nebo též $\mathcal{NP}$ obtížná. Poznamenejme, že to vlastně znamená, že $\mathcal{U}$ je alespoň tak těžká jako všechny $\mathcal{NP}$ úlohy.

1.9.7 Cookova věta. Úloha SAT, splňování formulí v konjunktivním normálním tvaru, je $\mathcal{NP}$ úplná úloha.

1.9.8 Myšlenka důkazu. Není těžké se přesvědiít, že úloha SAT je ve třídě $\mathcal{NP}$. První fáze nedeterministického algoritmu vygeneruje ohodnocení logických proměnných a na základě tohoto ohodnocení jsme schopni v polynomiálním čase ověřit, zda je v tomto ohodnocení formule pravdivá nebo ne.

Druhá část důkazu spočívá v popisu práce Turingova stroje formulí výrokové logiky. Načrtneme základní myšlenku tohoto popisu.

Je dán nedeterministický Turingův stroj M s množinou stavů Q, vstupní abecedou Σ, páskovou abecedou Γ, přechodovou funkcí δ, počátečním stavem q_0 a koncovým stavem q_f. Předpokládejme, že M příjímá slovo w a potřebuje přitom $p(n)$ kroků.

Zavedeme logické proměnné:

$h_{i,j}$, $i = 0, 1, \ldots, p(n)$, $j = 1, 2, \ldots, p(n)$;

fakt, že hodnota proměnné $h_{i,j}$ je rovna 1 znamená, že hlava Turingova stroje v čase i čte j-té pole pásky.

s_i^q, $i = 0, 1, \ldots, p(n)$, $q \in Q$;

fakt, že hodnota proměnné s_i^q je rovna 1 znamená, že Turingův stroj v čase i je ve stavu q.

$t_{i,j}^A$, $i = 0, 1, \ldots, p(n)$, $j = 1, 2, \ldots, p(n)$, $A \in \Gamma$;

fakt, že hodnota proměnné $t_{i,j}^A$ rovna 1 znamená, že v čase i v j-tém poli pásky je páskový symbol A.

Nyní je třeba formulemi popsat následující fakta:

1. V každém okamžiku je Turingův stroj v právě jednom stavu.
2. V každém okamžiku čte hlava Turingova stroje právě jedno pole vstupní pásky.
3. V každém okamžiku je na každém poli pásky Turingova stroje právě jeden páskový symbol.
4. Na začátku práce (tj. v čase 0) je Turingův stroj ve stavu q_0, hlava čte první pole pásky a na pásce je na prvních n polích vstupní slovo, ostatní pole pásky obsahují B.
5. Krok Turingova stroje je určen přechodovou funkcí, tj. stav stroje, obsah čteného pole a poloha hlavy v čase $i+1$ je dána přechodovou funkcí.
6. V polích pásky, které v čase i hlava nečte, je obsah v čase $i+1$ stejný jako v i.
7. Na konci práce Turingova stroje, tj. v čase $p(n)$, je stroj ve stavu q_f.

Ukážeme jak utvořit formule pro body 1, 4, 5, 6 a 7.

Bod 1. V okamžiku i je Turingův stroj v aspoň jednom stavu:

$$\bigvee_{q \in Q} s_i^q.$$

V okamžiku i Turingův stroj není ve dvou různých stavech:

$$\bigwedge_{q\neq q'} (\neg s_i^q \vee \neg s_i^{q'}).$$

Nyní fakt, že Turingův stroj je v okamžíku i právě jednom stavu je konjunkce obou výše uvedených formulí:

$$(\bigvee_{q\in Q} s_i^q) \wedge \bigwedge_{q\neq q'} (\neg s_i^q \vee \neg s_i^{q'}).$$

Bod 4. Na začátku práce (tj. v čase 0) je Turingův stroj ve stavu q_0, hlava čte první pole pásky a na pásce je na prvních n polích vstupní slovo $a_1 a_2 \dots a_n$, ostatní pole obsahují B.

$$s_0^{q_0} \wedge h_{0,1} \wedge t_{0,1}^{a_1} \wedge \dots \wedge t_{0,n}^{a_n} \wedge t_{0,n+1}^B \wedge \dots \wedge t_{0,p(n)}^B.$$

Bod 5. Jestliže Turingův stroj je v čase i ve stavu q, hlava je na j-tém poli pásky, hlava čte páskový symbol A a $\delta(q, A)$ se skládá z trojic (p, C, D) (zde $D = 1$ znamená posun hlavy doprava, $D = -1$ znamená posun hlavy doleva), pak formule má tvar:

$$\bigwedge_j \bigwedge_{A\in\Gamma} ((s_i^q \wedge h_{i,j} \wedge t_{i,j}^A) \Rightarrow \bigvee (s_{i+1}^p \wedge t_{i+1,j}^C \wedge h_{i+1,j+D})).$$

Bod 6. Obsah polí kromě j-tého zůstává v čase $i + 1$ stejný:

$$\bigwedge_j \bigwedge_{A\in\Gamma} ((\neg h_{i,j} \wedge t_{i,j}^A) \Rightarrow t_{i+1,j}^A).$$

Bod 7. Na konci práce Turingova stroje, tj. v čase $p(n)$ je stroj ve stavu q_f.

$$s_{p(n)}^{q_f}.$$

Výslednou formuli dostaneme jako konjunkci všech dílčích formulí pro všechny časové okamžiky $i = 0, 1, \dots, p(n)$.

1.10 Převody úloh

1.10.1 Na základě tvrzení 1.9.4 víme: K důkazu, že rozhodovací úloha $\mathcal{U}$ ze třídy $\mathcal{NP}$ je $\mathcal{NP}$ úplná stačí, abychom ukázali, že se na $\mathcal{U}$ polynomiálně redukuje některá $\mathcal{NP}$ úplná úloha. Zatím jediná $\mathcal{NP}$ úplná úloha, kterou známe, je SAT, splňování booleovských formulí v konjunktivním normálním tvaru. Ukážeme řadu polynomiálních redukcí a tím ukážeme, že i další rozhodovací úlohy jsou $\mathcal{NP}$ úplné.

1.10.2 $3-CNF\ SAT$**.** Úloha: Je dána formule φ v konjunktivním normálním tvaru, kde každá klasusule má 3 literály.
Otázka: Je formule φ splnitelná?

1.10.3 Tvrzení. Platí

$$SAT \lhd_p 3 - CNF\,SAT.$$

1.10.4 Nástin převodu SAT na $3 - CNF\ SAT$. Je dána formule φ v konjunktivním normálním tvaru. Zkonstruujeme formuli ψ, která

1. je v konjunktivním normálním tvaru, kde každá klausule obsahuje maximálně 3 literály;
2. je splnitelná právě tehdy, když je splnitelná formule φ.

Označme $C_1, C_2, \ldots, C_k$ všechny klausule formule φ. Jestliže každá z klausulí obsahuje nejvýše 3 literály, nemusíme nic konstruovat, v tomto případě je $\psi = \varphi$.

Pro každou klausuli C, která obsahuje víc než 3 literály, sestrojíme formuli ψ_C takto: Nechť $C = l_1 \vee l_2 \vee \ldots \vee l_s$, kde l_i jsou literály. Zavedeme nové logické proměnné $x_1, x_2, \ldots, x_{s-3}$ a položíme

$$\psi_C = (l_1 \vee l_2 \vee x_1) \wedge (\neg x_1 \vee l_3 \vee x_2) \wedge (\neg x_2 \vee l_4 \vee x_3) \wedge \ldots \wedge (\neg x_{s-3} \vee l_{s-1} \vee l_s).$$

Platí: Formule ψ_C je splnitelná iff C je splnitelná.

Formuli ψ dostaneme jako konjunkci všech klasusulí formule φ, které mají nejvýše 3 literály a formulí ψ_C pro klausule C o více než 3 literálech.

Předpokládejme, že formule φ má k klausulí a nejdelší klausule má s literálů. Pak v konstrukci ψ jsme přidali maximálně $(s-3)k$ nových logických proměnných (rovnost nastává v případě, že každá z klausulí formule φ obsahuje přesně $s > 3$ literálů). Navíc jsme formuli prodloužili o maximálně o $2(s-3)k$ literálů (každá nová logická proměnná se ve formuli ψ objevuje přesně dvakrát). Tedy délka formule ψ se pouze polynomiálně zvětšila vzhledem k délce formule φ.

1.10.5 Důsledek. Protože úloha $3 - CNF\ SAT$ je ve třídě $\mathcal{NP}$, jedná se o $\mathcal{NP}$ úplnou úlohu.

1.10.6 Obarvení vrcholů grafu. Je dán prostý neorientovaný graf bez smyček $G = (V, E)$. *obarvení vrcholů* grafu G je přiřazení, které každému vrcholu v grafu G přiřazuje jeho barvu $b(v)$, $b(v)$ je prvek množiny (barev) B, pro které platí, že žádné dva vrcholy spojené hranou nemají stejnou barvu. (Jinými slovy, jestliže $\{u, v\}$ je hrana grafu G, pak $b(u) \neq b(v)$.)

Graf G se nazývá k*-barevný*, jestliže jeho vrcholy je možné obarvit k barvami (tj. množina B má k prvků).

1.10.7 k-barevnost. Úloha: Je dán prostý neorientovaný graf G bez smyček a číslo k.
Otázka: Je graf G k-barevný?

1.10.8 Tvrzení. Platí

$$3 - CNF\ SAT \lhd_p \text{3-barevnost.}$$

1.10.9 Základní myšlenka převodu. Je dána formule φ, která je v CNF a každá klausule má 3 literály. K důkazu je třeba zkonstruovat prostý neorientovaný graf G bez smyček takový, že φ je splnitelná právě tehdy, když G je 3-barevný. Konstrukce využívá pomocný graf o pěti vcholech $\{1, 2, 3, 4, 5\}$ a pěti hranách s touto vlastností:

- Jestliže vrcholy 1 a 2 mají stejnou barvu, pak tuto barvu musí mít i vrchol 5.
- Jestliže jeden z vrcholů 1 a 2 má barvu z, pak lze tento graf obarvit tak, aby i vrchol 5 měl barvu z.

1.10.10 Důsledek. Protože 3-barevnost je ve třídě $\mathcal{NP}$, jedná se o $\mathcal{NP}$ úplnou úlohu.

1.10.11 Tvrzení. Platí

$$\text{3-barevnost } \lhd_p ILP.$$