Jméno: Noblesní Plšice **UČO:** 1234567

0007 *list* 1 *učo* 1234567 *body*

Oblast strojově snímaných informací. Své učo a číslo listu vyplňte zleva dle vzoru číslic. Jinak do této oblasti nezasahujte. 0123456789

2. [2,5 bodu] Nechť $\mathcal{A}$ je deterministický konečný automat (DFA) se vstupní abecedou Σ (připomeňme, že Σ je dána jakožto součást n-tice popisující $\mathcal{A}$). Řekneme, že $\mathcal{A}$ je *znakově univerzální* pokud existuje slovo $w \in \Sigma^*$ splňující tyto 3 podmínky:

- w je akceptováno automatem $\mathcal{A}$
- délka w je menší nebo rovna počtu stavů automatu $\mathcal{A}$
- každý znak abecedy Σ se ve w vyskytuje alespoň jednou (formálně, s využitím značení z FJA/Automatů: $\forall a \in \Sigma : \#_a(w) \geq 1$).

Dokažte, že problém

$$\textit{CHAR-UNIV} = \{\langle \mathcal{A} \rangle \mid \mathcal{A} \text{ je znakově univerzální DFA } \Sigma\}$$

je NP-úplný.

Příslušnost do NP. Problém *CHAR-UNIV* je rozhodován např. nedeterministickým Turingovým strojem, který pracuje následovně. Stroj nejprve deterministicky ověří, zda vstup je kódem nějakého DFA $\mathcal{A}$ (to lze jistě učinit v polynomiálním čase). Pokud tomu tak není, ihned zamítne. V opačném případě uhádne nějaké slovo w délky nejvýše n (kde n je počet stavů automatu $\mathcal{A}$) nad abecedou Σ automatu $\mathcal{A}$. Následně deterministicky ověří, zda w obsahuje každé písmeno abecedy Σ. Pokud tomu tak není, stroj ihned zamítne. V opačném případě stroj deterministicky odsimuluje výpočet $\mathcal{A}$ nad w a ověří, že tento výpočet končí v akceptujícím stavu (pokud ano, pak stroj akceptuje, jinak zamítne). Všechny uvedené výpočty jsou proveditelné nedeterministickýcm Turingovým strojem, který běží v polynomiálním čase.

NP-těžkost. Ukážeme, že *3-SAT* $\leq_p$ *CHAR-UNIV*.

Redukci f definujeme jako

$$f(w) = \begin{cases} \langle \mathcal{A}_\emptyset \rangle & \text{pokud } w \text{ není kódem výrokové formule v 3cnf,} \\ \langle \mathcal{A}_\varphi \rangle & \text{pokud } w \text{ je kódem výrokové formule } \varphi \text{ v 3cnf,} \end{cases}$$

kde $\mathcal{A}_\emptyset$ je nějaký fixní DFA bez akceptujících stavů a $\mathcal{A}_\varphi$ je DFA zkonstruovaný v závislosti na výrokové formuli φ v 3cnf následovně. Nechť $\varphi = C_1 \wedge \ldots \wedge C_m$ ($C_1, \ldots, C_m$ jsou klauzule φ) a nechť φ obsahuje výrokové proměnné $x_1, \ldots, x_n$. Automat $\mathcal{A}_\varphi = (Q_\varphi, \Sigma_\varphi, \delta_\varphi, q_0, F_\varphi)$ konstruujeme takto:

- Do množiny stavů Q_φ nejprve přidáme stavy $q_0, q_0', q_1, q_2, \ldots, q_{n+1}$. Další stavy budeme do Q_φ přidávat v průběhu konstrukce.
- Položíme $\Sigma_\varphi = \{1, \ldots, m, t, f\}$, tedy máme jeden znak pro každou klauzuli, který odpovídá číslu klauzule, a dále pomocné znaky t a f.
- Automat má jediný koncový stav $F = \{q_{n+1}\}$.
- Zbývá definovat přechodovou funkci. V následujícím budeme psát $p \xrightarrow{a} q$ namísto $\delta_\varphi(p, a) = q$ a tento zápis přechodů budeme pro větší přehlednost řetězit. Stavy q_0, q_0' mají po jednom přechodu: $q_0 \xrightarrow{t} q_0' \xrightarrow{f} q_1$. Pro každou výrokovou proměnnou x_i provedeme následující:
  - Vypočteme množiny $\mathit{pos}(i) = \{j \mid 1 \leq j \leq m \text{ a } C_j \text{ obsahuje literál } x_i\}$ a $\mathit{neg}(i) = \{j \mid 1 \leq j \leq m \text{ a } C_j \text{ obsahuje literál } \neg x_i\}$.
  - Označme $k = |\mathit{pos}(i)|$. Pokud $k = 0$, přidáme přechod $q_i \xrightarrow{t} q_{i+1}$.

- Pokud $k > 0$, můžeme psát $pos(i) = \{j_1, \ldots, j_k\}$. Přidáme do Q_φ nové stavy $p_i^1, p_i^2, \ldots, p_i^k$ a definujeme přechody $q_i \xrightarrow{t} p_i^1 \xrightarrow{j_1} p_i^2 \xrightarrow{j_2} \ldots \xrightarrow{j_{k-1}} p_i^k \xrightarrow{j_k} q_{i+1}$. Tedy vytvoříme v $\mathcal{A}_\varphi$ „cestu“ z q_i do q_{i+1}, přičemž na této cestě $\mathcal{A}_\varphi$ čte symbol t a následně symboly, které odpovídají klauzulím obsahujícím x_i.
- Označme $\ell = |neg(i)|$. Pokud $\ell = 0$, přidáme přechod $q_i \xrightarrow{f} q_{i+1}$.
- Pokud $\ell > 0$, můžeme psát $neg(i) = \{j_1, \ldots, j_\ell\}$. Přidáme do Q_φ nové stavy $s_i^1, s_i^2, \ldots, s_i^\ell$ a definujeme přechody $q_i \xrightarrow{f} s_i^1 \xrightarrow{j_1} s_i^2 \xrightarrow{j_2} \ldots \xrightarrow{j_{\ell-1}} s_i^\ell \xrightarrow{j_\ell} q_{i+1}$. Tedy vytvoříme v $\mathcal{A}_\varphi$ „cestu“ z q_i do q_{i+1}, přičemž na této cestě $\mathcal{A}_\varphi$ čte symbol f a následně symboly, které odpovídají klauzulím obsahujícím $\neg x_i$.

Snadno lze ověřit, že výše uvedená konstrukce skutečně vytvoří DFA. Navíc snadno vytvoříme program s polynomiální časovou složitostí ($\mathcal{O}(m \cdot n)$), který ji provádí, a tento program lze implementovat deterministickým Turingovým strojem s polynomiální časovou složitostí.

Zbývá tedy ukázat, že $w \in$ *3-SAT* $\iff$ $f(w) \in$ *CHAR-UNIV*. Tato ekvivalence zřejmě platí pro w, která nejsou kódy 3cnf formulí. Ve zbytku důkazu tedy předpokládejme, že $w = \langle\varphi\rangle$, kde φ je formule v 3cnf. Před samotným důkazem učiňme následující pozorování: slova akceptovaná automatem $\mathcal{A}_\varphi$ jsou právě slova tvaru $tfy_1u_1 \cdots y_nu_n$, kde $y_i \in \{t, f\}$ a kde složení podslov u_i je dáno tím, zda $y_i = t$ či $y_i = f$. V prvním případě je u_i tvořeno právě znaky z $pos(i)$, ve druhém případě je u_i tvořeno právě znaky z $neg(i)$. (Všimněme si, že u_i mohou být prázdná.) Navíc zřejmě každé slovo akceptované tímto automatem má délku menší, než je počet stavů tohoto automatu.

"$\Rightarrow$" Nechť $w = \langle\varphi\rangle \in$ *3-SAT*. Pak existuje přiřazení ν proměnným ve φ takové, že $\nu(\varphi) = \mathit{true}$. Uvažme slovo $u = tfy_1u_1 \cdots y_nu_n$ takové, že $y_i = t$ pokud $\nu(x_i) = \mathit{true}$ a $y_i = f$ pokud $\nu(x_i) = \mathit{false}$. Podslova $u_1, \ldots, u_n$ jsou dána předpisem výše. Protože ν je splňující přiřazení, každá klauzule C_j obsahuje literál vyhodnocený na true. Pokud je tento literál tvaru x_i pro nějaké i, pak $j \in pos(i)$. Zároveň $\nu(x_i) = \mathit{true}$ a tedy u_i (a tudíž i u) obsahuje všechny znaky z $pos(i)$, včetně j. Podobně pokud je splněný literál v C_j tvaru $\neg x_i$ pro nějaké i, pak $j \in neg(i)$ a zároveň $\nu(x_i) = \mathit{false}$, tedy u_i (a u) obsahují j. Protože C_j bylo zvoleno libovolně, dostáváme že u obsahuje všechna $1 \le j \le m$. Zřejmě též u obsahuje t a f a je akceptováno automatem $\mathcal{A}_\varphi$. Tedy $f(w) = \langle\mathcal{A}_\varphi\rangle \in$ *CHAR-UNIV*.

"$\Leftarrow$" Nechť u je slovo akceptované automatem $\mathcal{A}_\varphi$, které obsahuje každý znak abecedy alespoň jednou. Z výše uvedeného plyne, že $u = tfy_1u_1 \cdots y_nu_n$ kde $y_i \in \{t, f\}$. Uvažme přiřazení ν takové, že pro $1 \le i \le n$ je $\nu(x_i) = \mathit{true}$ pokud $y_i = t$, jinak $\nu(x_i) = \mathit{false}$. Ukážeme, že $\nu(\varphi) = \mathit{true}$. Nechť C_j je libovolná klauzule φ. Víme, že znak j se vyskytuje někde v u. Existuje tedy $1 \le i \le n$ takové, že u_i obsahuje j. Pokud $y_i = t$, pak u_i obsahuje právě znaky z $pos(i)$ a tedy $x_i \in C_j$; zároveň z definice ν je $\nu(x_i) = \mathit{true}$, tedy C_j je splněna. Podobně pokud $y_i = f$, pak u_i obsahuje právě znaky z $neg(i)$ a tedy $\neg x_i \in C_j$; zároveň z definice ν je $\nu(x_i) = \mathit{false}$ a tedy $\nu(\neg x_i) = \mathit{true}$. I v tomto případě je tedy C_j je splněna. Jelikož ν splňuje všechny klauzule z φ, je φ splnitelná a $w = \langle\varphi\rangle \in$ *3-SAT*.