

KMI/AKTI 2025/2026

Petr Osička

Disclaimer: Poznámky k semináři. Úkoly jsou na konci textu.

Obsah

1	Synchronizační slova	2
1.1	Délka nejkratšího synchronizačního slova	3
1.2	Obtížnost nalezení nejkratšího synchronizačního slova	5
2	Univerzalita	7
3	Úkoly	8

1 Synchronizační slova

Uvažujeme deterministický konečný automat bez počátečního a akceptujících stavů (dále jen automat), který je dán

- konečnou množinou stavů Q ,
- konečnou množinou akcí Σ , (někdy jí říkáme abeceda),
- přechodovou funkcí $\delta : Q \times \Sigma \rightarrow Q$.

Přechodovou funkci přirozeně rozšíříme na řetězce a množiny stavů. (O které rozšíření se jedná poznáme podle typu argumentů). Pracujeme s konečnými slovy nad Σ , prázdné slovo značíme ϵ .

$$\begin{aligned}\delta(q, \epsilon) &= q \\ \delta(q, xa) &= \delta(\delta(q, x), a) \text{ pro } q \in Q, \text{ slovo } x \text{ a akci } a \\ \delta(S, w) &= \{\delta(q, w) \mid q \in S\} \text{ pro } S \subseteq Q \text{ a slovo } w\end{aligned}$$

Řekneme, že slovo w *synchronizuje*, pokud $|\delta(Q, w)| = 1$.

Na semináři jsme si ukázali příklady synchronizujících a nesynchronizujících slov v konkrétním automatu. Ukázali jsme si i příklad automatu, pro který žádné synchronizační slovo neexistuje.

Budeme se zabývat následujícím algoritmickým problémem: Je dán automat $\mathcal{A} = (Q, \Sigma, \delta)$. Existuje k němu synchronizační slovo? (V případě, že ano, řekneme, že $\mathcal{A}$ synchronizuje.)

Naivní algoritmus funguje následovně. Sestavíme podmnožinový automat $2^{\mathcal{A}} = (2^Q, \Sigma, \delta')$, přechodová funkce δ' je zde rozšířenou přechodovou funkcí pro δ . Pokud v sestaveném automatu najdeme slovo vedoucí ze stavu Q do nějakého jednoprvkového stavu, existuje k $\mathcal{A}$ synchronizační slovo.

Na semináři jsme si ukázali příklad běhu algoritmu na jednoduchém automatu. Diskutovali jsme i zdánlivě efektivnější variantu, kdy začneme od Q a budujeme $2^{\mathcal{A}}$ na požádání. I s tímto vylepšením má ovšem algoritmus exponenciální složitost vzhledem k $|Q|$, protože se může stát, že algoritmus sestaví exponenciálně velkou část automatu $2^{\mathcal{A}}$.

Předchozí algoritmus se ovšem dá vylepšit tak, že sestaví pouze polynomicky velkou část $2^{\mathcal{A}}$. Řekneme, že w je *slévací slovo* pro stavy s, t , pokud $\delta(s, w) = \delta(t, w)$. Pokud automat synchronizuje, je příslušné synchronizační slovo slévacím slovem pro libovolnou dvojici stavů. Naopak, pokud pro každou dvojici stavů existuje slévací slovo, automat synchronizuje. Synchronizující slovo najdeme pomocí následujícího algoritmu.

1. Nastavíme $x \leftarrow \epsilon$.

2. Dokud $|\delta(Q, x)| > 1$

- vybereme dva různé stavy $s, t \in \delta(Q, x)$, označme jejich slévací slovo w
- nastavíme $x \leftarrow xw$

3. Vratíme x .

Z definice slévacího slova plyne, že po každé iteraci cyklu se velikost množiny $\delta(Q, x)$ zmenší alespoň o jedna. Dojde tedy k situaci, kdy je $\delta(Q, x)$ jednoprvková a x je synchronizujícím slovem.

Pokud existuje dvojice stavů s, t , pro kterou slévací slovo neexistuje, automat očividně nesynchronizuje: pro libovolné slovo w máme $|\delta(\{s, t\}, w)| = 2$ a proto $|\delta(Q, w)| \geq 2$.

Slévací slova můžeme hledat vytvořením automatu (Q', Σ, δ') , kde Q' je množina všech jednoprvkových a dvouprvkových podmnožin Q , a δ' je rozšířenou přechodovou funkcí pro δ . Pokud v tomto automatu najdeme cestu ze stavu $\{s, t\}$ do jednoprvkového stavu, existuje pro dvojici s, t slévací slovo, které odečteme z hran na cestě. (Cestu můžeme nalézt například průchodem do šířky).

Algoritmus pro nalezení synchronizujícího slova za předpokladu existence slévacích slov tedy snadno rozšíříme na algoritmus, který slévací slova sám hledá. Pokud se mu slévací slovo nalézt nepodaří, vrátí informaci o tom, že automat nesynchronizuje.

Vzhledem k $|Q|$ má automat (Q', Σ, δ') kvadratický počet stavů. Délka jednotlivých slévacích slov je tak také omezena kvadraticky. Algoritmus má $|Q| - 1$ iterací, během jedné iterace provede nejvýše kubický počet operací (při výpočtu $\delta(Q, x)$; v algoritmu si mezi iteracemi pamatujeme $\delta(Q, x)$: na začátku nastavíme $C \leftarrow Q$ a v každé iteraci potom pouze $C \leftarrow \delta(C, w)$).

1.1 Délka nejkratšího synchronizačního slova

Označme $n = |Q|$. Počet dvojic stavů je $n(n - 1)/2$, toto číslo tak omezuje i délku slévacího slova. Synchronizující slovo nalezené algoritmem z předchozí kapitoly tak má délku nejvýše $n(n - 1)^2/2$. Na délku nejkratšího synchronizujícího slova existují i lepší horní odhady, např. $(n^3 - n)/6$.

Nyní si ukážeme, že existuje rodina automatů, ve kterých je délka nejkratšího synchronizujícího slova rovna $(n - 1)^2$. Jedná se tzv. *Černého automaty*.

Černého automat s n stavy je automat s množinou stavů $\{0, 1, \dots, n - 1\}$, abecedou $\{a, b\}$ a přechodovou funkcí danou:

$$\begin{aligned}
\delta(0, a) &= 1, \\
\delta(0, b) &= 1, \\
\delta(i, b) &= i + 1 \pmod{n}, \text{ pro } i > 1, \\
\delta(i, a) &= i, \text{ pro } i > 1
\end{aligned}$$

Automat je synchronizován slovem $(ab^{n-1})^{n-2}a$, jehož délka je $n(n-2) + 1 = (n-1)^2$. Že slovo skutečně synchronizuje, je vidět z následujících pozorování.

- Pro $k > 1$ máme

$$\begin{aligned}
\delta(\{0, \dots, k\}, a) &= \{1, \dots, k\} \\
\delta(\{1, \dots, k\}, b^{n-1}) &= \{0, \dots, k-1\}
\end{aligned}$$

- Trik z předchozího bodu zopakujeme $n-2$ krát a dostaneme $\{0, 1\}$. Nyní si všimneme, že $\delta(\{0, 1\}, a) = \{1\}$.

Věta 1: Černého

Každé synchronizující slovo v Černého automatu s n stavy má délku nejméně $(n-1)^2$.

Důkaz. Nechť w je nejkratší synchronizační slovo. Toto slovo jistě nekončí b , protože to by šlo odebrat a slovo by pořád synchronizovalo. Můžete tedy psát $w = w'a$ a máme $\delta(Q, w') = \{0, 1\}$.

Uvnitř w' je každé a následováno b . Pro libovolnou množinu stavů P totiž platí $\delta(P, a) = \delta(P, aa)$ (první a odebere z P stav 0, pokud tam je, a další provedení akce a již množinu nezmění).

Zavede nový symbol c a řekneme, že $c = ab$. Ve slově w' nahradíme všechny výskyty slova ab symbolem c , dostaneme tak slovo v nad abecedou $\{b, c\}$. Z Černého automatu odstraníme symbol a a nahradíme jej c , dostaneme automat $\mathcal{B}$

$$\begin{aligned}
\delta'(0, b) &= 1, \\
\delta'(0, c) &= 2, \\
\delta'(i, c) &= i + 1 \pmod{n} \text{ pro } i > 0.
\end{aligned}$$

Z konstrukce plyne, že v automatu $\mathcal{B}$ máme $\delta'(Q, v) = \{0, 1\}$ a slovo vc tak synchronizuje do stavu 2. Pro každé slovo $u \in \{b, c\}^*$ tak máme, že uvc synchronizuje do stavu 2. Pro $l > |vc|$ tak platí, že pro každý stav i existuje

v (grafu) automatu $\mathcal{B}$ tah délky l ze stavu i do stavu 2 , pokud zvolíme $i = 2$, je tah uzavřený. Každý uzavřený tah v $\mathcal{B}$ se skládá z kružnic délek n a $n - 1$, tj. máme

$$l = k_1 n + k_2 (n - 1),$$

kde $k_1, k_2 \in \mathbb{N}_0$ (přirozená čísla s 0). Nyní použijeme lemma z teorie čísel.

Lemma: Pro nesoudělná $p, q \in \mathbb{N}$ je největší číslo r takové, že neexistují $k_1, k_2 \in \mathbb{N}_0$ tak, že $r = k_1 p + k_2 q$ rovno $pq - p - q$.

Protože čísla n a $n - 1$ jsou nesoudělná, předchozí lemma implikuje

$$|vc| > n(n - 1) - n - (n - 1) = n^2 - 3n + 1.$$

Protože $\delta'(1, b) = 2$ a $\delta'(1, c) = 2$, provedení vc ze stavu 1 by vyžadovalo existenci uzavřeného tahu (s počátkem ve stavu 2), máme dokonce

$$|vc| \geq n^2 - 3n + 2.$$

Protože vc synchronizuje $\mathcal{B}$ musí obsahovat alespoň $n - 1$ symbolů c , a v tedy obsahuje alespoň $n - 2$ symbolů c . Máme tak (protože $c = ab$)

$$\begin{aligned} |w'| &\geq n^2 - 3n + 2 + n - 2 = n^2 - 2n, \\ |w'a| &\geq n^2 - 2n + 1 = (n - 1)^2 \end{aligned}$$

Otevřený problém: Dokažte nebo vyvráťte *Černého domněnku*: Délka nejkratšího synchronizujícího slova v libovolném automatu s n stavy je nejvýše $n - 2$.

□

1.2 Obtížnost nalezení nejkratšího synchronizačního slova

Vysvětlili jsme si, že stačí ukázat NP-obtížnost rozhodovací verze, tj. problému jehož vstupem jsou synchronizující automat $\mathcal{A}$ a přirozené číslo k ; očekáváme odpověď ano, pokud pro $\mathcal{A}$ existuje synchronizační slovo délky nejvýše k .

NP-obtížnost ukážeme redukcí z problému 3SAT: splnitelnosti booleovských formulí v konjunktivní normální formě, jejíž klausule mají právě tři literály.

Nechť φ je formule s proměnnými $x_1, \dots, x_n$ a klauzulemi $C_1, \dots, C_m$. Vytvoříme automat $\mathcal{A}$ s množinou stavů

$$\{s\} \cup \bigcup_{i=1}^m \{s_1^i, s_2^i, \dots, s_{n+1}^i\}.$$

(Horní index odpovídá indexu klausule, dolní index, mimo $n + 1$, indexu proměnné.) Uvážíme abecedu $1, 0$ (pravda, nepravda). Pro $i = 1, 2, \dots, m$ a $j = 1, 2, \dots, n$ dodefinujeme potřebnou část přechodové funkce Pokud $x_j \in C_i$

$$\begin{aligned}\delta(s_j^i, 1) &= s, \text{ zkratka} \\ \delta(s_j^i, 0) &= s_{j+1}^i.\end{aligned}$$

Pokud $\neg x_j \in C_i$

$$\begin{aligned}\delta(s_j^i, 1) &= s_{j+1}^i, \\ \delta(s_j^i, 0) &= s. \text{ zkratka}\end{aligned}$$

Jinak

$$\begin{aligned}\delta(s_j^i, 1) &= s_{j+1}^i, \\ \delta(s_j^i, 0) &= s_{j+1}^i.\end{aligned}$$

Nakonec dodáme pro $a = 0, 1$

$$\begin{aligned}\delta(s_{n+1}^i, a) &= S \\ \delta(s, a) &= S\end{aligned}$$

Za číslo k zvolíme n .

Vidíme, že automat má $m(n + 1) + 1$ stavů a lze jej vytvořit v polynomickém čase.

Pokud je φ pravdivá pro ohodnocení proměnných $e : \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$, pak je $e(x_1)e(x_2)\dots e(x_n)$ synchronizační slovo délky k .

Nechť je $\mathcal{A}$ synchronizován slovem $w = w_1 \dots w_n$ délky n (kratší nemůže být, protože synchronizovat se dá pouze do s). Přitom w_i bereme jako ohodnocení proměnné x_i . Protože w synchronizuje, pro každé $1 \leq i \leq m$ existuje j tak, že z s_j^i jsme šli zkratkou do s (Z S_i^i se tam totiž bez zkratky dostaneme do s až po $n + 1$ krocích). To ovšem znamená, že literál proměnné x_j obsažený v C_i je ohodnocen pravdivě a C_i je také pravdivý. Ohodnocení dané w tak formuli splní.

Na semináři jsme si ukázali příklad redukce.

Z výše uvedené obtížnosti tedy plyne, že pokud $P \neq NP$, nelze nalézt minimální synchronizační slovo v polynomickém čase.

2 Univerzalita

Viz dodané slajdy a článek [6], který lze dohledat na arxiv.org.

3 Úkoly

Vypracované a v latexu vysázené ukoly mi pošlete mailem a poté přijďte osobně vysvětlit. Obrázky lze namalovat ručně.

1. (**Synchronizační slova.**) Zafixujeme abecedu $\Sigma = \{0, 1\}$. Množina $X \subseteq \Sigma^*$ je prefixový kód, pokud pro všechny $x, y \in X$ platí, že x není prefixem y . Prefixový kód X je maximální, pokud neexistuje prefixový kód Y tak, že $X \subset Y$. Slovo $w \in X$, kde X je maximální prefixový kód, je synchronizující, pokud pro každé slovo $v \in \Sigma^*$ platí, že $wv \in X^*$. (X^* zde značí uzávěr jazyka X , tedy jazyk obsahující slova vzniklá zřetěžením konečného počtu slov z jazyka X .)

Ke konečnému prefixovému kódu X můžeme sestavit automat A_X , tak, že $L(A_X) - \{\epsilon\} = X$. Konstrukce je následující:

- Množina stavů Q je tvořena všemi vlastními prefixy prvků X (w je vlastní prefix w' , pokud je jeho prefixem a $w \neq w'$), včetně prázdného slova ϵ .
- přechodová relace $\Delta \subseteq Q \times \Sigma \times Q$ je dána
 - $(q, a, qa) \in \Delta$ pokud je qa vlastní prefix nějakého slova z X ,
 - $(q, a, \epsilon) \in \Delta$ pokud $qa \in X$
- počáteční a akceptující stav je ϵ .

Znění úkolů

- a) Lze z nějaké vlastnosti A_X poznat, zda-li je X maximální?
 - b) Předpokládejme, že X je maximální. Dokažte, že w je synchronizující pro X , právě když je w synchronizační slovo v A_X . Do kterého stavu w synchronizuje?
2. (**Univerzalita.**) Sestavte automat, na který získáme popsanou redukcí z formule

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4) \wedge (x_1 \vee \neg x_3 \vee \neg x_4)$$

Reference

- [1] Sven Sandberg. Homing and synchronizing sequences. Model Based Testing of Reactive Systems. LNCS 3472, 2005.
- [2] B.K. Natarajan. Algorithmic approach to the automated design of part orienters. SFCS 1986.
- [3] David Eppstein. Reset sequences for monotonic automata. SIAM Journal of Computing, 19(3), 1990.
- [4] Černý Ján. Poznámka k homogénym experimentom s konečnými automatmi. Matematicko-Fyzikálny časopis Slovenskej Akadémie Vied, 14, 1964.
- [5] M.V. Volkov Synchronizing automata and the Černý conjecture. LNCS 5196, 2008.
- [6] Jiří Balun, Tomáš Masopust, Petr Osička. Speed Me up If You Can: Conditional Lower Bounds on Opacity Verification. MFCS 2023.