Univerzita Pavla Jozefa Šafárika v Košiciach
Prírodovedecká fakulta

OPERÁCIA ŠTVOREC NA JAZYKOCH REPREZENTOVANÝCH DETERMINISTICKÝMI, ALTERNUJÚCIMI A BOOLEOVSKÝMI AUTOMATMI

DIPLOMOVÁ PRÁCA

Študijný odbor: **Informatika**
Školiace pracovisko: **Ústav informatiky**
Vedúci záverečnej práce: **RNDr. Galina Jirásková, CSc.**

Košice 2016 **Bc. Ivana Krajňáková**

Poďakovanie

Rada by som poďakovala vedúcej diplomovej práce RNDr. Galine Jiráskovej, CSc. za cenné pripomienky a za obetavosť počas tvorby mojej diplomovej práce.

Namiesto tejto strany vložte zadanie z informačného systému podpísané vedúcim ústavu!

Abstrakt

V práci študujeme operáciu štvorec na deterministických, alternujúcich a booleovských konečnostavových automatoch. Najskôr sa venujeme tesnosti horného odhadu $(n-k)\cdot 2^n + k\cdot 2^{n-1}$ pre zložitosť štvorca jazyka reprezentovaného minimálnym deterministickým n stavovým automatom na binárnej abecede, v ktorom je k koncových stavov, kde $1 \leq k \leq n-2$. Pre každé takéto n a k popíšeme binárny jazyk akceptovaný n stavovým deterministickým automatom s k koncovými stavmi taký, že minimálny deterministický automat pre jeho štvorec má $(n-k)\cdot 2^n + k\cdot 2^{n-1}$ stavov. Preskúmame aj prípad štvorca na jazykoch reprezentovaných deterministickými automatmi, v ktorých iba jeden stav je nekoncový. Tam dostávame horný odhad $(n+2)\cdot 2^{n-2}$ v prípade, že počiatočný stav je koncový, a horný odhad $(n+3)\cdot 2^{n-2}$, ak počiatočný stav je nekoncový. V obidvoch prípadoch dokážeme tesnosť týchto odhadov. Na dôkaz tesnosti používame v prvom prípade binárnu abecedu a ternárnu v druhom. Zároveň v druhom prípade nájdeme binárny jazyk, ktorého štvorec dosahuje hodnotu $(n+3)\cdot 2^{n-2} - 1$.

Ďalej zhrnieme už známe poznatky ohľadom booleovských a alternujúcich automatov. Budeme sa venovať hornému odhadu $2^m + n + 1$ stavovej zložitosti zreťazenia kvôli uvedeniu otvoreného problému tesnosti tohto odhadu, ktorý formulovali Fellah, Jürgensen, Yu [1990, Internat. J. Computer Math. 35, 117–132]. Využitím nášho jazyka, ktorý je ťažký pre štvorec na deterministických automatoch definujeme taký binárny jazyk akceptovaný n stavovým alternujúcim automatom, že každý alternujúci automat pre jeho štvorec má aspoň $2^n + n + 1$ stavov. Zovšeobecnením tohto nášho výsledku ukážeme tesnosť horného odhadu $2^m + n + 1$ pre zložitosť zreťazenia jazykov reprezentovaných alternujúcimi automatmi s m a n stavmi. Týmto vyriešime už spomínaný otvorený problém z roku 1990.

Kľúčové slová: regulárne jazyky, operácia štvorec, deterministické, alternujúce a booleovské konečnostavové automaty, stavová zložitosť

Abstract

We study the square operation on languages represented by deterministic, alternating, and boolean finite automata. First, we examine the tightness of the upper bound $(n-k)\cdot 2^n + k\cdot 2^{n-1}$ on the state complexity of the square on binary languages represented by a minimal deterministic automaton with n states and k final states, where $1 \leq k \leq n-2$. For every such n and k, we describe a binary language accepted by an n state deterministic automaton with k final states such that the minimal deterministic automaton for its square has $(n-k)\cdot 2^n + k\cdot 2^{n-1}$ states. Then we investigate the square of languages accepted by deterministic automata with only one nonfinal state. We obtain two results: the tight upper bound $(n+2)\cdot 2^{n-2}$ on the state complexity of the square of languages represented by binary deterministic automata with $n-1$ final states, where the initial state is final; and the tight upper bound $(n+3)\cdot 2^{n-2}$ on the state complexity of the square of languages accepted by ternary deterministic automata, where the initial state is the only nonfinal state. The bound $(n+3)\cdot 2^{n-2} - 1$ is reachable with binary alphabet.

In the second part of our thesis, we summarize known results on boolean and alternating automata. We explain the upper bounds $2^m + n + 1$ for the concatenation to be able to formulate an open problem of its tightness stated by Fellah, Jürgensen, Yu [1990, Internat. J. Computer Math. 35, 117–132]. Using our binary language that is hard for the square operation on deterministic automata we define a binary language accepted by alternating automaton with n states such that every alternating automaton for its squate has at least $2^n + n + 1$ states. By generalizing this result we are able to prove the tightness of the upper bound 2^m+n+1 for the concatenation of languages represented by alternating automata with m and n states. This resolves the mentioned open problem from 1990.

Keywords: regular languages, square operation, deterministic, alternating, and boolean finite automata, state complexity

Obsah

Úvod

Štvorec je základná unárna operácia na formálnych jazykoch definovaná ako $L^2 = \{uv \mid u \in L, v \in L\}$. Každý regulárny jazyk vieme charakterizovať jeho stavovou zložitosťou, čo predstavuje počet stavov v minimálnom deterministickom automate, ktorý akceptuje tento jazyk. Je známe, že ak jazyk L je akceptovaný n stavovým deterministickým konečnostavovým automatom, tak potom jazyk L^2 je akceptovaný deterministickým automatom, ktorý má najviac $n \cdot 2^n - 2^{n-1}$ stavov [9].

Toto horné ohraničenie je odvodené z horného ohraničenia $m \cdot 2^n - 2^{n-1}$ stavovej zložitosti zreťazenia $KL = \{uv \mid u \in K \text{ a } v \in L\}$ jazykov K a L, ktoré sú akceptované m a n stavovými deterministickými automatmi [8, 11]. Yu et al. [11] taktiež ukázali, že horné ohraničeni $m \cdot 2^n - 2^{n-1}$ pre zreťazenie sa nedosahuje, ak prvý jazyk v zreťazení je akceptovaný deterministickým automatom s viac ako jedným koncovým stavom. V takom prípade ukázali horné ohraničenie $(m-k) \cdot 2^n + k \cdot 2^{n-1}$, kde prvý jazyk v zreťazení je reprezentovaný m stavovým deterministickým automatom s k koncovými stavmi a minimálny deterministický automat pre druhý jazyk má n stavov. Toto horné ohraničenie je tesné pre každé k také, že $1 \leq k \leq n-1$ na každej abecede, ktorá obsahuje aspoň dva znaky [4].

Tieto výsledky boli použité na definovanie jazykov, ktoré by boli ťažké pre zreťazenie aj na alternujúcich automatoch, na ktorých je známe horného ohraničenie $2^m + n + 1$ [3]. Jeho tesnosť formulovali autori z [3] ako otvorený problém, ktorý bol takmer vyriešený v [5], kde bola dosiahnutá stavová zložitosť zreťazenia $2^m + n$. Jazyky použité v [5] boli akceptované 2^m a 2^n stavovými deterministickými automatmi, kde v obidvoch automatoch bola polovica stavov koncová. Avšak zreťazenie týchto jazykov nedosahuje horné ohraničenie stavovej zložitosti pre zreťazenie, preto tieto jazyky nie sú vhodnými svedkami.

Našou motiváciou je rovnaký problém štvorca na alternujúcich automatoch, preto v tejto práci študujeme túto operáciu detailnejšie. Z práce Rampersada [9] je známy binárny jazyk akceptovaný n stavovým deterministickým automatom s jedným koncovým stavom taký, že minimálny determinis-

tický automat pre jeho štvorec má $n \cdot 2^n - 2^{n-1}$ stavov. Ak minimálny automat pre jazyk L má k koncových stavov, kde $k > 1$, tak zložitosť jeho štvorca nedosahuje uvedenú hornú hranicu. V takom prípade je zložitosť jazyka L^2 najviac $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ [11].

V našej predošlej práci [2] sme našli ternárny jazyk akceptovaný n stavovým automaton s k koncovými stavmi, ktorého štvorec dosahuje horné ohraničenie zložitosti pre túto operáciu. Ukazáli sme tak tesnosť tohto ohraničenia, avšak až od ternárnej abecedy. Naše výpočty z tejto práce však naznačili, že existuje jazyk akceptovaný n stavovým deterministickým automatom s k koncovými stavmi, ktorého štvorec dosahuje horné ohraničenie už na binárnej abecede. Ďalej sme z výpočtov zistili, že na binárnej ani ternárnej abecede sa ohraničenie $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ nedosahuje, ak pôvodný jazyk bol akceptovaný automatom s $n-1$ koncovými stavmi.

V prvej časti práce sa budeme venovať tesnosti horného odhadu zložitosti $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ pre štvorec jazyka, ktorý bol pôvodne akceptovaný n stavovým deterministickým automatom a mal k koncových stavov. Pre každé n a k také, že $1 \leq k \leq n-2$, nájdeme binárny jazyk akceptovaný n stavovým deterministickým automatom s k koncovými stavmi taký, že počet stavov minimálneho deterministickeho automatu pre jeho štvorec je $(n-k) \cdot 2^n + k \cdot 2^{n-1}$. Toto je hlavný výsledok našej práce, ktorý neskôr použijeme pri skúmaní štvorca na alternujúcich a booleovských automatoch.

Následne preštudujeme stavovú zložitosť štvorca jazykov, ktoré sú reprezentované n stavovými deterministickými automatmi s jediným nekoncovým stavom. Rozlíšime dva prípady. Najskôr preskúmame, čo sa deje v prípade, ak počiatočný stav patril medzi koncové stavy. Vtedy ukážeme horný odhad $(n+2) \cdot 2^{n-2}$, ktorý bude tesný na binárnej abecede. Potom preskúmame prípad, keď len počiatočný stav je nekoncovým stavom. Vtedy dostávame horný odhad $(n+3) \cdot 2^{n-2}$ a dokážeme, že je tesný na ternárnej abecede. Na binárnej abecede sa dopracujeme k hranici $(n+3) \cdot 2^{n-2} - 1$, ktorá je len o jedna menšia od maximálnej možnej zložitosti v tomto prípade.

V druhej časti práce sa budeme venovať booleovským a alternujúcim automatom. Zhrnieme základné poznatky o nich a ukážeme, že ak jazyk L je akceptovaný n stavovým alternujúcim automatom, potom jazyk L^2 je akceptovaný alternujúcim s najviac $2^n + n + 1$ stavmi [3]. Náš automat použitý na dôkaz tesnosti pre odhad $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ použijeme znova pri definovaní binárneho jazyka, ktorý už bude akceptovaný n stavovým alternujúcim automatom a pomocou tohto jazyka ukážeme tesnosť horného odhadu $2^n + n + 1$ stavovej zložitosti štvorca akceptovaného alternujúcim automatom. Zovšeobecnením potom vyriešime už spomínaný otvorený problém tesnosti horného odhadu zložitosti zreťazenia z [3]. Nakoniec ešte ukážeme, že zložitosť operácie štvorec na booleovských automatoch je $2^n + n$.

Kapitola 1

Definície a základné pojmy

Pod *abecedou* rozumieme ľubovoľnú konečnú množinu. Jej prvky nazývame *znaky* (symboly). *Slovo* v danej abecede Σ je ľubovoľná a konečná postupnosť symbolov abecedy Σ. Množinu všetkých slov v danej abecede označíme Σ^*. *Jazyk* nad danou abecedou Σ je ľubovoľná podmnožina Σ^*.

Definujme *deterministický konečnostavoý automat (deterministic finite automaton, DFA)* ako usporiadanú päticu $M = (Q, \Sigma, \delta, s, F)$, kde Q je konečná množina stavov, Σ je vstupná abeceda, $\delta : Q \times \Sigma \to Q$ je prechodová funkcia, $s \in Q$ je počiatočný stav, $F \subseteq Q$ je množina akceptujúcich stavov. Prechodovú funkciu vieme rozšíriť na $\delta : Q \times \Sigma^* \to Q$ rekurzívne: $\delta(q, \varepsilon) = q$, pre všetky $q \in Q$; a $\delta(q, wa) = \delta(\delta(q, w), a)$, pre všetky stavy $q \in Q$, všetky slová $w \in \Sigma^*$ a pre každé písmeno $a \in \Sigma$. Pod jazykom akceptovaným automatom M rozumieme množinu slov takú, že $L(M) = \{w \in \Sigma^* \mid \delta(s, w) \in F\}$.

Stav q v automate M nazývame *dosiahnuteľný*, ak existuje $w \in \Sigma^*$ také, že $\delta(s, w) = q$. Stav q je nedosiahnuteľný, ak nie je dosiahnuteľný. V automate $M = (Q, \Sigma, \delta, s, F)$ považujeme stavy p, q za *ekvivalentné*, $p \sim q$, ak platí $\delta(p, w) \in F$ práve vtedy, keď $\delta(q, w) \in F$ pre všetky slová $w \in \Sigma^*$. Automaty M a M' považujeme za *ekvivalentné*, ak $L(M) = L(M')$.

Automat M je *minimálny*, ak každý s ním ekvivalentný automat má aspoň toľko stavov ako M. Je známe, že DFA je minimálny, ak spĺňa nasledujúce podmienky: Všetky jeho stavy sú dosiahnuteľné a žiadne dva stavy nie sú ekvivalentné. Pod *stavovou zložitosťou (state complexity)* jazyka L, označujeme $\mathrm{sc}(L)$, rozumieme počet stavov minimálneho DFA pre jazyk L.

Nedeterministický konečnostavový automat (nondeterministic finite automaton, NFA) N je usporiadaná pätica $N = (Q, \Sigma, \delta, I, F)$, kde Q je konečná množina stavov, Σ je vstupná abeceda, $\delta : Q \times \Sigma \to 2^Q$ je prechodová funkcia (2^Q označuje množinu všetkých podmnožín Q), $I \subseteq Q$ je množina počiatočných stavov a $F \subseteq Q$ je množina koncových stavov. Analogicky rozšírime prechodovú funkciu $\delta : 2^Q \times \Sigma^* \to 2^Q$ rekurzívne: $\delta(R, \varepsilon) = R$, pre všetky

$R \subseteq Q$; $\delta(R, wa) = \bigcup_{q \in \delta(R,w)} \delta(q, a)$, pre všetky množiny $R \subseteq Q$, všetky slová $w \in \Sigma^*$ a pre každé písmeno $a \in \Sigma$. Potom jazyk akceptovaný NFA N je množina slov $L(N) = \{w \in \Sigma^* : \delta(I, w) \cap F \neq \emptyset\}$.

Ku každému NFA $N = (Q, \Sigma, \delta, I, F)$ existuje s ním ekvivalentný DFA $M = (Q', \Sigma, \delta', s, F')$, ktorý možno získať tzv. *podmnožinovou konštrukciou* [10] nasledovne: Za stavy zoberieme všetky podmnožiny Q, $Q' = 2^Q$. Prechodovú funkciu definujeme ako $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$, pre všetky $R \in 2^Q$ a pre všetky $a \in \Sigma$. Počiatočným stavom s sa stane množina $I, s = I$. Množinu koncových stavov budú tvoriť tie podmnožiny, ktorých aspoň jeden prvok patrí v N medzi koncové stavy, $F' = \{R \in 2^Q \mid R \cap F = \emptyset\}$. Automat M nazývame podmnožinovým automatom k NFA N. Tento automat nemusí byť minimálnym, lebo niektoré jeho stavy môžu byť nedosiahnuteľné alebo ekvivalentné.

Lemma 1.1 *Nech $N = (Q, \Sigma, \delta, I, F)$ je NFA taký, že ku každému stavu $q \in Q$ existuje slovo w_q, ktoré je akceptované iba zo stavu q, t. j.:*
(a) $\delta(q, w) \cap F \neq \emptyset$;
(b) ak $p \neq q$, tak $\delta(p, w) \cap F = \emptyset$.
Potom podmnožinový automat k NFA N nemá ekvivalentné stavy.

Dôkaz.
Ak S a T sú dve rôzne podmnožiny Q, tak existuje stav q taký, že bez ujmy na všeobecnosti $q \in S$ ale $q \notin T$. Potom slovo w_q je v podmnožinovom automate akceptované zo stavu S a zamietnuté zo stavu T.

□

Pod *zreťazením* dvoch jazykov K a L rozumieme jazyk, ktorého slová vieme rozdeliť na dve časti, kde prvá je z jazyka K a druhá z L. Formálne $KL = \{uv \mid u \in K, v \in L\}$. Pod *štvorcom* jazyka L rozumieme zreťazenie jazyka samého so sebou, teda $L^2 = LL$.

Zrkadlový obraz w^R (reverse) slova w je definovaný indukciou na dĺžku slova takto: $\varepsilon^R = \varepsilon$ a ak $w = av$, pre $a \in \Sigma$ a $v \in \Sigma^*$, tak $w^R = av^R$. Zrkadlový obraz jazyka L je jazyk $L^R = \{w^R \mid w \in L\}$.

1.1 Konštrukcia NFA pre štvorec

V tejto časti uvedieme postup, pomocou ktorého vyrobíme z DFA A pre L nový automat N akceptujúci už jazyk L^2. Tento automat bude síce najskôr nedeterministický, ale pomocou podmnožinovej konštrukcie vieme ľahko vytvoriť z NFA N pre L^2 deterministický automat M pre L^2.

Majme teda DFA $A = (Q, \Sigma, \delta, s, F)$ pre jazyk L, s n stavmi, ktoré označíme $q_0, q_1, \ldots, q_{n-1}$, pričom $s = q_0$. Potom nedeterministický konečnostavový automat $N = (Q', \Sigma, \delta', S', F')$ pre jazyk L^2 skonštruujeme takto:

Na vytvorenie N použijeme dve kópie automatu A. Stavy v prvej kópii ponecháme v pôvodnom značení, v druhej len prečíslujeme na $0, \ldots, n-1$. Teda Q' vyzerá nasledovne $Q' = \{q_0, \ldots, q_{n-1}, 0, \ldots, n-1\}$. Počiatočným stavom N bude $S' = \{q_0\}$, ak $q_0 \notin F$, inak N bude mať dva počiatočné stavy, q_0 a 0, $S' = \{q_0, 0\}$. Koncovými stavmi N sa stanú koncové stavy druhej kópie A. Inými slovami, ak $q_i \in F$ pre nejaké $i \in \{0, \ldots, n-1\}$, tak $i \in F'$. Všetky stavy z prvej kópie $\{q_0, \ldots, q_{n-1}\}$ teda budú nekoncové.

Zostáva nám ešte definovať prechodovú funkciu. Prechody v rámci kópií zostávanú nezmenené, avšak po dočítaní časti slova v prvej kópií potrebujeme dočítať slovo v druhej kópii. Prechodovú funkciu δ' definujeme nasledovne:

- V druhej kópii nemeníme nič $\delta'(i, a) = \{j\}$, ak $\delta(q_i, a) = q_j$.
- V prvej kópii pridáme prechod do počiatočného stavu druhej kópie, ak sme v stave ktorý bol pôvodne koncovým.

Formálne teda pre i, j z množiny $\{0, 1, \ldots, n-1\}$ máme:

$$\delta'(q_i, a) = \begin{cases} \{\delta(q_i, a)\}, & \text{ak } \delta(q_i, a) \notin F; \\ \{\delta(q_i, a), 0\}, & \text{ak } \delta(q_i, a) \in F; \end{cases}$$

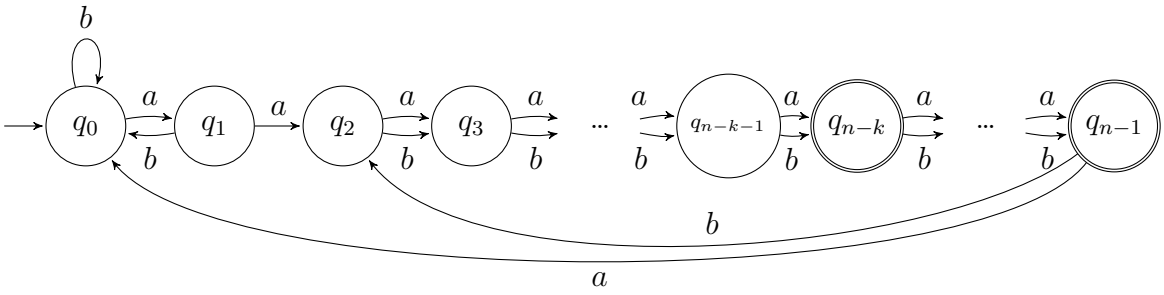
$$\delta'(i, a) \quad = \{j\}, \text{ kde } j \text{ je také, že } \delta(q_i, a) = q_j.$$

Teraz už vieme skonštruovať nedeterministický automat pre štvorec ľubovoľného jazyka L akceptovaný n stavovým DFA. Túto konštrukciu budeme ďalej využívať v nasledujúcich dôkazoch pri vytváraní NFA pre štvorec konkrétnych jazykov.

Kapitola 2

Štvorec na deterministických automatoch

V tejto kapitole sa budeme venovať horným odhadom stavovej zložitosti štvorca jazykov, ktoré sú reprezentované deterministickými s viacerými koncovými stavmi. Je známe, že ak L je akceptovaný n stavovým DFA s k koncovými stavmi, tak potom L^2 je akceptovaný DFA s najviac $(n-k)\cdot 2^n + k\cdot 2^{n-1}$ stavmi [11]. V práci [2] sme pre každé n a k také, že $1 \leq k \leq n-2$ našli ternárny jazyk, ktorého štvorec dosahuje túto hodnotu. V prvej časti tejto kapitoly tento výsledok vylepšíme použitím binárneho jazyka. V druhej časti rozoberieme otvorený prípad horného odhadu zložitosti štvorca, ak pôvodný jazyk má $n-1$ koncových stavov.

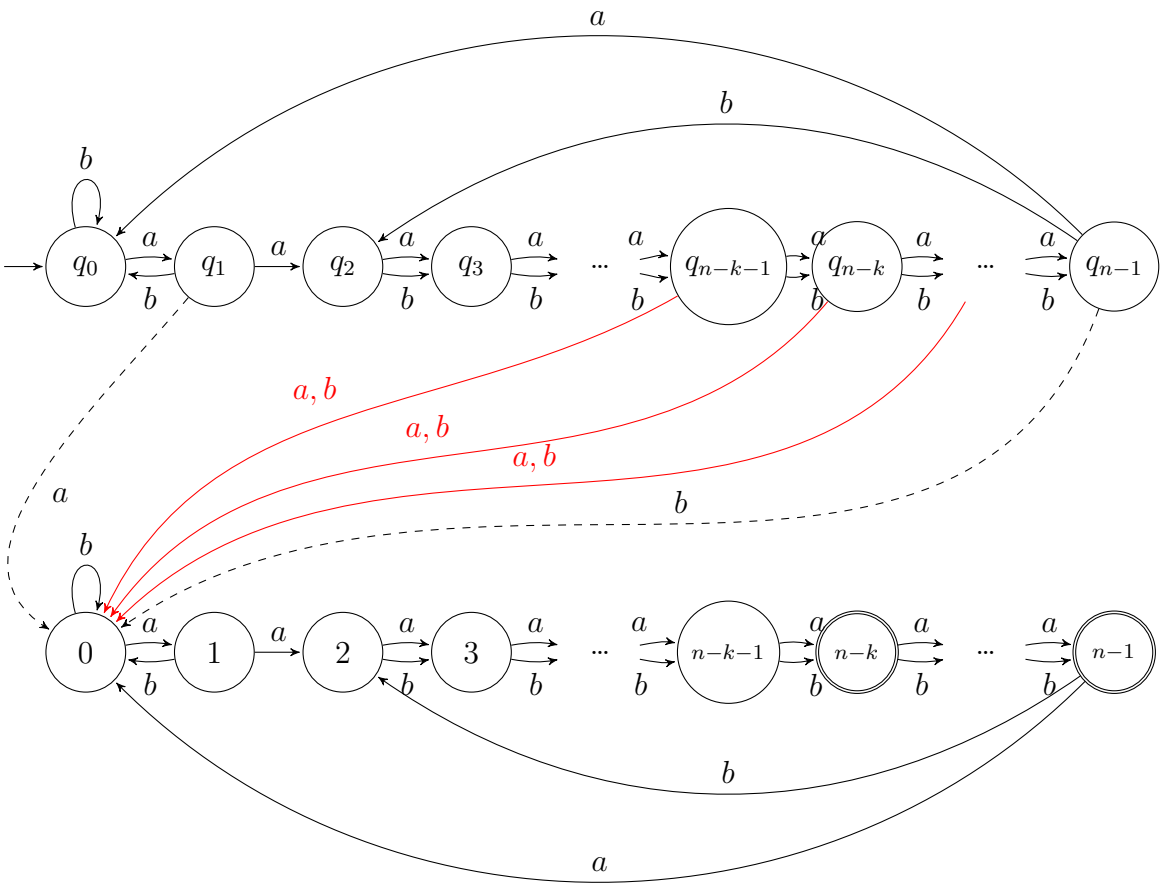


Obr. 2.1: DFA A s k koncovými stavmi taký, že $\mathrm{sc}(L(A)^2) = (n-k)2^n + k2^{n-1}$.

Lemma 2.1 *Nech $n \geq 3$, $1 \leq k \leq n-2$ a nech L je jazyk nad abecedou Σ taký, že $\mathrm{sc}(L) = n$ a minimálny DFA pre L má k koncových stavov. Potom $\mathrm{sc}(L^2) \leq (n-k)\cdot 2^n + k\cdot 2^{n-1}$ a táto hranica je tesná, ak $|\Sigma| \geq 2$.*

Dôkaz.
Zo spôsobu vytvorenia NFA pre štvorec podľa konštrukcie v stati 1.1 a následnej podmnožinovej konštrukcie vieme, že sú dosiahnuteľné len množiny dvoch typov: $\{q_i\} \cup S; S \subseteq \{0, 1, 2, \ldots, n-1\}$, $0 \leq i \leq n-k-1$, takýchto podmnožín je spolu $(n-k) \cdot 2^n$; $\{q_i, 0\} \cup S; n-k \leq i \leq n-1; S \subseteq \{1, 2, 3, \ldots, n-1\}$, takýchto podmnožín je spolu $k \cdot 2^{n-1}$. Priestor týchto množín budeme označovať ako $\mathcal{R}$. Vidíme, že v $\mathcal{R}$ je presne $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ množín. Tým sme ukázali horný odhad.

Pre tesnosť zoberme jazyk L akceptovaný DFA A podľa obrázka 2.1. Zostrojme NFA N pre jazyk L^2. Pre korektný dôkaz potrebujeme ukázať, že podmnožinový automat M k NFA N má $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ dosiahnuteľných stavov, ktoré sú navzájom neekvivalentné. Začnime s dosiahnuteľnosťou.



Obr. 2.2: NFA N pre L^2, čiarkované šípky sa doplnia len v prípade ak $k = n-2$.

Matematickou indukciou podľa veľkosti množín teraz ukážeme, že v M sú všetky množiny z $\mathcal{R}$ dosiahnuteľné. Jedno- a dvojprvkové množiny sú dosiahnuteľné pretože v podmnožinovom automate máme:

$$\to \{q_0\} \xrightarrow{a} \{q_1\} \xrightarrow{a} \cdots \xrightarrow{a} \{q_{n-k-1}\} \xrightarrow{a} \{q_{n-k}, 0\},$$

$$\{q_{n-k}, 0\} \xrightarrow{b} \{q_{n-k+1}, 0\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_{n-2}, 0\} \xrightarrow{b} \{q_{n-1}, 0\},$$

$$\{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\}, \quad \{q_0, 1\} \xrightarrow{b} \{q_0, 0\},$$

$$\{q_0, 1\} \xrightarrow{a} \{q_1, 2\} \xrightarrow{b} \{q_0, 3\} \xrightarrow{b} \{q_0, 4\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_0, n-1\} \xrightarrow{b} \{q_0, 2\},$$

$$\{q_0, j-i\} \xrightarrow{a^i} \{q_i, j\}, \text{pre } i = 0, 1, \ldots, n-k-1, \text{ a } j = 0, 1, \ldots, n-1.$$

Predpokladajme, že všetky t prvkové množiny z $\mathcal{R}$ sú dosiahnuteľné. Ukážeme, že potom sú všetky $t+1$ prvkové množiny z $\mathcal{R}$ dosiahnuteľné. Nech $S = \{q_i, s_1, s_2, \ldots, s_t\}$, kde $q_i \in Q$, a $0 \leq s_1 < s_2 < \cdots < s_t \leq n-1$ je $t+1$ prvková množina z $\mathcal{R}$. Rozlíšime niekoľko prípadov.

(1) Nech $i = n-k$ a keďže $S \in \mathcal{R}$, tak $s_1 = 0$. Zoberme t prvkovú množinu z $\mathcal{R}$, $\{q_{n-k-1}, s_2 - 1, s_3 - 1, \ldots, s_t - 1\}$, ktorá je podľa indukčného predpokladu dosiahnuteľná. Potom:

$$\{q_{n-k-1}, s_2 - 1, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_{n-k}, 0, s_2, s_3, \ldots s_t\} = S.$$

Všimnime si, že z pôvodne t prvkovej, nám vznikla už $t+1$ prvková množina.

(2) Nech $n-k \leq i \leq n-1$. Indukciou podľa i ukážeme, že každá $t+1$ prvková množina $\{q_i, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$ je dosiahnuteľná. Prvý krok indukcie, teda $i = n-k$, sme ukázali v bode (1). Predpokladajme, že každá $t+1$ prvková množina $\{q_i, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$, kde $n-k \leq i \leq n-2$, je dosiahnuteľná. Ukážeme, že potom aj každá $t+1$ prvková množina $\{q_{i+1}, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$ je dosiahnuteľná. Nech $S = \{q_{i+1}, 0, s_2, \ldots, s_t\}$. Rozlíšime dva prípady:

(2a) Nech $s_2 = 1$. Potom máme

$$\{q_i, 0, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_{i+1}, 0, 1, s_3, \ldots, s_t\} = S.$$

Množina vľavo je dosiahnuteľná podľa indukčného predpokladu z indukcie podľa t, má t prvkov, obsahuje 0, teda je z $\mathcal{R}$.

(2b) Nech $s_2 \geq 2$. Potom

$$\{q_i, 0, \delta(s_2, b^{n-3}), \ldots, \delta(s_t, b^{n-3})\} \xrightarrow{b} \{q_{i+1}, 0, s_2, s_3, \ldots, s_t\} = S.$$

Pritom množina vľavo je dosiahnuteľná podľa indukčného predpokladu pre i.

Poznámka Keďže prechody na b tvoria cyklus dĺžky $n-2$ na prvkoch 2 a viac, takže $\delta(j, b^{n-3})$ je vlastne $\delta(j, b^{-1})$. To znamená, že $\delta(j, b^{n-3})$ nás posunie zo stavu j do stavu $j-1$.

(3) Nech $i = 0$. Rozoberieme niekoľko prípadov.

(3a) Ak $s_1 = 0$, $s_2 = 1$, potom

$$\{q_{n-1}, 0, s_3 - 1, \ldots, s_t - 1, n-1\} \xrightarrow{a} \{q_0, 0, 1, s_3, \ldots, s_t\} = S.$$

Množina vľavo je $t+1$ prvková a je dosiahnuteľná podľa bodov (1) a (2).

(3b) Ak $s_1 = 1$, $s_2 \geq 2$, potom

$$\{q_{n-1}, 0, s_2 - 1, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_0, 1, s_2, \ldots, s_t\} = S.$$

Množina vľavo je $t+1$ prvková a je dosiahnuteľná podľa bodov (1) a (2).

(3c) Ak $s_1 = 0$, $s_2 \geq 2$, potom

$$\{q_0, 1, \delta(s_2, b^{n-3}), \ldots, \delta(s_t, b^{n-3})\} \xrightarrow{b} \{q_0, 0, s_2, \ldots, s_t\} = S.$$

Množina vľavo je dosiahnuteľná podľa (3b).

(3d) Ak $s_1 = 2$ ($s_2 \geq 3$), potom

$$\{q_0, 1, s_2 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_1, 2, s_2, \ldots, s_t\} \xrightarrow{b^{n-2}} \{q_0, 2, s_2, \ldots, s_t\} = S.$$

Množina vľavo je dosiahnuteľná podľa (3b).

(3e) Ak $s_1 \geq 3$, potom podľa bodu (3d)

$$\{q_0, 2, s_2 - s_1 + 2, \ldots, s_t - s_1 + 2\} \xrightarrow{b^{s_1-2}} \{q_0, s_1, s_2, \ldots, s_t\} = S.$$

(4) Nech $1 \leq i \leq n-k-1$, potom

$$\{q_0, s_1 - i, \ldots, s_t - i\} \xrightarrow{a^i} \{q_i, s_1, \ldots, s_t\} = S.$$

Množina vľavo je dosiahnuteľná podľa (3). Tým je dokázaná dosiahnuteľnosť.

Teraz ukážeme rozlíšiteľnosť stavov v podmnožinovom automate M. Zoberme teraz dva rôzne stavy $p = \{q_i\} \cup S$ a $q = \{q_j\} \cup T$ z $\mathcal{R}$. Ukážme najskôr, že slovo $w = b(ab^{n-2})^{n-3}$ je v NFA N akceptované len zo stavu $n-1$. Keďže

$$\{n-1\} \xrightarrow{b} \{2\} \xrightarrow{a} \{3\} \xrightarrow{b^{n-2}} \{3\} \xrightarrow{ab^{n-2}} \{4\} \xrightarrow{ab^{n-2}} \{5\} \xrightarrow{ab^{n-2}} \ldots \xrightarrow{ab^{n-2}} \{n-1\},$$

slovo w je akceptované zo stavu $n-1$. Ak by sme však začali z iného stavu t, $2 \leq t \leq n-2$, tak potom $\delta(t, b) \in \{3, 4, \ldots, n-1\}$, odkiaľ každý stav sa dostane do $\{0\}$ na slovo $(ab^{n-2})^{n-3}$. Podobne, $\{0, 1\}$ na slovo w prechádza do

$\{0\}$. Pre stavy z pôvodnej kópie máme $\delta(\{q_i\}, w) \subseteq \{q_j, 0\}$, kde $0 \leq i \leq n-1$ a pre j platí, že buď $j = 0$, ak $i < n-1$, alebo $j = n-1$, ak $i = n-1$. Ďalej si všimnime, že slovo $a^{n-1-t}w$ je v N akceptované len zo stavu t, $0 \leq t \leq n-2$. Z toho nám vyplýva, že stavy p a q sú rozlíšiteľné, ak $S \neq T$.

Uvažujme teraz prípad keď $S = T$. Máme teda $\{q_i\} \cup S$, $\{q_j\} \cup S$, kde $0 \leq i < j \leq n-1$ a $S \subseteq \{0, 1, \ldots, n-1\}$. Zaveďme označenie $\underline{x}$ znamenajúce, že x môže alebo nemusí byť v množine, v závislosti od počtu koncových stavov. Rozoberme prípady:

(1) Nech $i = 0$ a $j = 1$. Potom

$$\{q_0\} \cup S \xrightarrow{(ab^{n-2})^{n-2}} \{q_0, 0, \underline{n-1}\} \xrightarrow{a} \{q_1, \underline{0}, 1\} \xrightarrow{a^{n-k-1}} \{q_{n-k}, 0, n-k-1, n-k\},$$

$$\{q_1\} \cup S \xrightarrow{(ab^{n-2})^{n-2}} \{q_{n-1}, 0, \underline{n-1}\} \xrightarrow{a} \{q_0, \underline{0}, 1\} \xrightarrow{a^{n-k-1}} \{q_{n-k-1}, n-k-1, n-k\}.$$

Takúto dvojicu stavov vieme rozlíšiť, lebo sa líšia v stave z druhej kópie.

(2) Nech $i = 0$ a $j \geq 2$. Potom

$$\{q_0\} \cup S \xrightarrow{b^{n-1-j}} \{q_0\} \cup S_1 \xrightarrow{a} \{q_1\} \cup S_2,$$

$$\{q_j\} \cup S \xrightarrow{b^{n-1-j}} \{q_{n-1}\} \cup S_1' \xrightarrow{a} \{q_0\} \cup S_2'.$$

Ak sú množiny S_1 a S_1', respektíve S_2 a S_2', rovnaké, tak pokračujeme ako v prípade 3a. Ak sú rôzne, tak pokračujeme ako v prípade keď $S \neq T$.

(3) Nech $i \geq 1$ a $i \leq j$.

$$\{q_i\} \cup S \xrightarrow{a^{n-1-j}} \{q_x\} \cup S_1 \xrightarrow{a} \{q_{x+1}\} \cup S_2, \quad x < n-1,$$

$$\{q_j\} \cup S \xrightarrow{a^{n-1-j}} \{q_{n-1}\} \cup S_1' \xrightarrow{a} \{q_0\} \cup S_2'.$$

Ak sú množiny S_1 a S_1', respektíve S_2 a S_2', rovnaké, tak pokračujeme ako v prípade 3a. Ak sú rôzne, tak pokračujeme ako v prípade keď $S \neq T$.

□

Ukázali sme tesnosť horného odhadu $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ pre štvorec. Týmto sme zlepšili výsledok z [2] z roku 2014, kde bola nutná ternárna abeceda. Tento dôkaz však nepokrýva stavovú zložitosť štvorca v prípade jediného nekoncového stavu v minimálnom DFA pre L.

2.1 Štvorec a jeden nekoncový stav

Rozobrali sme už zložitosť štvorca pre k koncových stavov. V predpokladoch lemy 2.1 sme však počet koncových stavov deterministického automatu pre jazyk L ohraničili na najviac $n-2$. Keď máme DFA pre L s $n-1$ koncovými stavmi, tak zo vzťahu pre stavovú zložitosť štvorca dostávame, že DFA pre L^2 môže mať najviac $(n-n+1)\cdot 2^n+(n-1)\cdot 2^{n-1}=(2n+2)\cdot 2^{n-2}$ stavov. Naše výpočty na zoznamoch neizomorfných automatov však ukázali, že táto hranica sa nedosahuje. Vo výpočtoch sme našli automaty (Obr. 2.3, 2.5) s $n-1$ koncovými stavmi, ktorých štvorec bol najťažší spomedzi ostatných binárnych automatov s $n-1$ koncovými stavmi.

Problém zložitosti štvorca jazyka akceptovaného DFA s $n-1$ koncovými stavmi sme rozdelili na dve časti, podľa toho, či počiatočný stav patrí medzi pôvodne koncové stavy alebo nie. V prvej časti sa zaoberáme prípadom, keď počiatočný stav je koncovým a dostávame tesný horný odhad $(n+2)\cdot 2^{n-2}$ zložitosti štvorca v tomto prípade. Na ukázanie tesnosti použijeme binárny automat z Obr. 2.3. Môžeme si všimnúť, že jeho prechody sú rovnaké ako pri automate na Obr. 2.1, no všetky jeho stavy sú koncové, okrem stavu q_1.

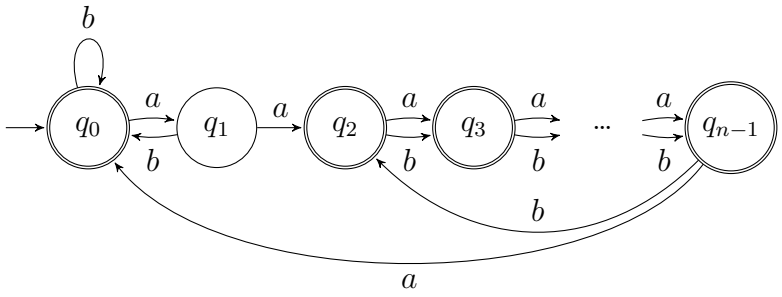
V druhej časti sa zaoberáme prípadom, keď počiatočný stav je jediným nekoncovým stavom v DFA pre jazyk L. V tomto prípade ukážeme, že horný odhad pre zložitosť štvorca je $(n+3)\cdot 2^{n-2}$. Pre dôkaz tesnosti tohto horného odhadu využijeme DFA na Obr. 2.5. Na konci tejto kapitoly zhrnieme naše výsledky a získame tesný horný odhad $(n+3)\cdot 2^{n-2}$ zložitosti štvorca, ak pôvodný jazyk bol akceptovaný deterministickým automatom s jediným nekoncovým stavom. Navyše pre binárnu abecedu dostane iba o jednotku menší dolný odhad. Vidíme teda, že hranica $(2n+2)\cdot 2^{n-2}$ sa v tomto prípade vôbec nedosahuje.

2.1.1 Zložitosť štvorca ak $|F|=n-1$ a $q_0\in F$

Rozoberme teraz situáciu, keď jazyk L je akceptovaný DFA s $n-1$ koncovými stavmi, a vieme, že počiatočný stav v DFA je koncový. V nasledujúcej leme ukážeme horný odhad zložitosti štvorca v tomto prípade a jeho tesnosť.

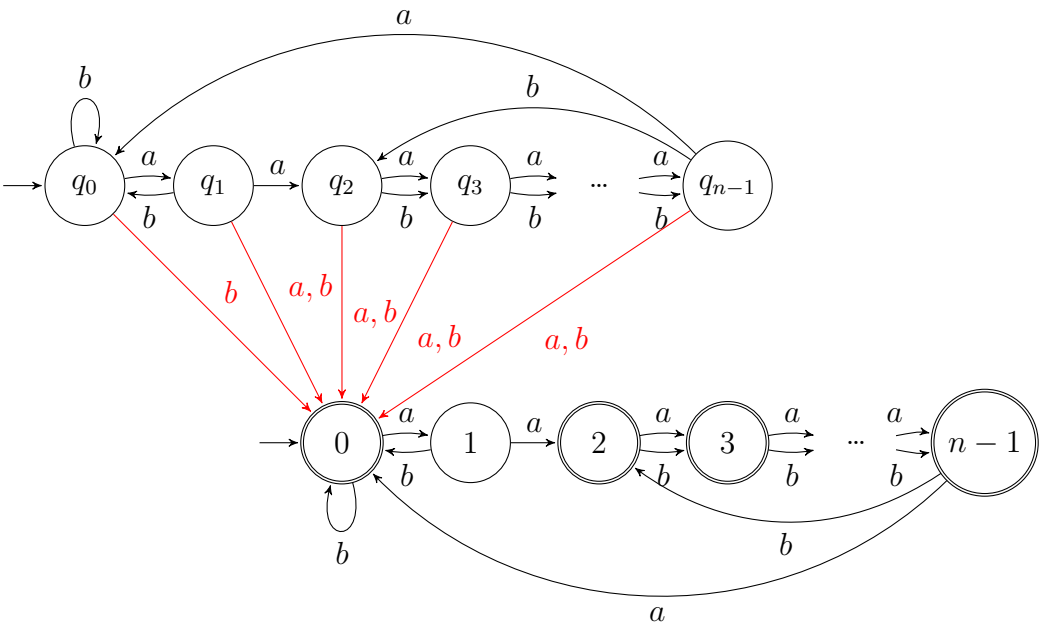
Lemma 2.2 *Nech $n\geq 3$ a nech L je regulárny jazyk nad abecedou Σ taký, že $\mathrm{sc}(L)=n$ a minimálny DFA pre L má $n-1$ koncových stavov a $q_0\in F$. Potom $\mathrm{sc}(L^2)\leq (n+2)\cdot 2^{n-2}$ a táto hranica je tesná, ak $|\Sigma|\geq 2$.*

Dôkaz.
Zo spôsobu vytvorenia NFA pre štvorec podľa konštrukcie v stati 1.1 a následnej podmnožinovej konštrukcie vieme, že sú dosiahnuteľné len tieto mno-



Obr. 2.3: DFA A s $n-1$ koncovými stavmi taký,že $\mathrm{sc}(L(A)^2) = (n+2)2^{n-2}$.

žiny: $\{q_0, 0\} \cup S; S \subseteq \{1, 2, \ldots, n-1\}$; $\{q_1, 1\} \cup S; S \subseteq \{0, 2, 3, \ldots, n-1\}$; $\{q_i, 0, i\} \cup S; 2 \leq i \leq n-1; S \subseteq \{1, 2, \ldots, n-1\} \setminus \{i\}$. Priestor týchto množín budeme označovať ako $\mathcal{R}$. Vidíme, že v $\mathcal{R}$ je presne $2^{n-1}+2^{n-1}+(n-2)\cdot 2^{n-2} = 2^n + (n-2)\cdot 2^{n-2} = (n+2)\cdot 2^{n-2}$ množín. Z toho vyplýva horný odhad.



Obr. 2.4: NFA N pre L^2.

Na dôkaz tesnosti zoberme DFA A podľa Obr. 2.3, ktorý bude akceptovať jazyk L. Zostrojme NFA N pre jazyk L^2. Pre korektný dôkaz potrebujeme ukázať, že podmnožinový automat M k NFA N má $2^n + (n-2)\cdot 2^{n-2}$ dosiahnuteľných stavov, ktoré sú navzájom neekvivalentné.

Matematickou indukciou podľa veľkosti množín ukážeme, že v M sú

všetky množiny z $\mathcal{R}$ dosiahnuteľné. Najmenej prvkové množiny v M sú dvojprvkové a tie sú dosiahnuteľné, keďže $\to \{q_0, 0\} \xrightarrow{a} \{q_1, 1\}$. Predpokladajme, že všetky t prvkové množiny z $\mathcal{R}$ sú dosiahnuteľné. Ukážeme, že potom sú všetky $t+1$ prvkové množiny z $\mathcal{R}$ dosiahnuteľné. Nech $S = \{q_i, s_1, s_2, \ldots, s_t\}$, kde $q_i \in Q$ je $t+1$ prvková množina z $\mathcal{R}$. V prípade, že S je prvého typu, pre jej prvky platí $0 \leq s_1 < s_2 < \cdots < s_t \leq n-1$. V prípade, že S je druhého typu, pre jej prvky platí $s_1 = 0$ a $s_0 < s_2 < \cdots < s_t \leq n-1$. V prípade, že S je tretieho typu, pre jej prvky platí $s_1 = 0$, $s_2 = i$ a $0 < s_3 < \cdots < s_t \leq n-1$. Rozlíšime niekoľko prípadov.

(1) Nech $i = 2$ a $s_1 = 0$ a $s_2 = i = 2$. Potom $\{q_1, 1, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_2, 0, 2, s_3, \ldots, s_t\}$. Množina vľavo je t prvková a teda dosiahnuteľná podľa indukčného predpokladu.

(2) Nech $i \geq 3$ a teda $s_1 = 0$ a $s_2 = i$. Potom dostávame $\{q_2, 0, 2, \delta(s_3, b^{n-2-i+2}), \ldots, \delta(s_t, b^{n-2-i+2})\} \xrightarrow{b^{i-2}} \{q_i, 0, i, s_3, \ldots, s_t\}$. Množinu vľavo sme vyrobili v bode (1).

Poznámka Keďže prechody na b tvoria cyklus dĺžky $n-2$ na prvkoch 2 a viac, takže $\delta(j, b^{n-2-i+2})$ je vlastne $\delta(j, b^{-i+2})$. To znamená, že $\delta(j, b^{n-3})$ nás posunie zo stavu j do stavu $j - i + 2$.

(3) Nech $i = 0$ a $s_2 = 1$. Potom $\{q_{n-1}, 0, n-1, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_0, 0, 1, s_3, \ldots, s_t\}$. Množinu vľavo sme vyrobili v bode (2).

(4) Nech $i = 0$ a $s_2 = 3$. Potom $\{q_0, 0, 1, s_3 - 2, \ldots, s_t - 2\} \xrightarrow{a}$ $\{q_1, 1, 2, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{b} \{q_0, 0, 3, s_3, \ldots, s_t\}$. Množinu vľavo sme vyrobili v bode (3).

(5) Nech $i = 0$ a $3 \leq s_2 \leq n-1$. Indukciou podľa s_2 ukážeme, že každá $t+1$ prvková množina $\{q_0, 0, s_2, \ldots, s_t\}$ z množiny $\mathcal{R}$ je dosiahnuteľná. Prvý krok indukcie, teda $s_2 = 3$, sme ukázali v bode (4). Predpokladajme, že každá $t+1$ prvková množina $\{q_0, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$, kde $3 \leq s_2 \leq n-2$, je dosiahnuteľná. Ukážeme, že potom je dosiahnuteľná aj každá $t+1$ prvková množina $\{q_0, 0, s_2 + 1, \ldots, s_t\}$ z $\mathcal{R}$. Nech $S = \{q_0, 0, s_2 + 1, s_3, \ldots, s_t\}$. Potom $\{q_0, 0, s_2, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{b} \{q_0, 0, s_2 + 1, s_3, \ldots, s_t\} = S$. Množina vľavo je dosiahnuteľná podľa indukcie podľa s_2.

(6) Nech $i = 0$ a $s_2 = 2$, $s_3 \geq 4$. Potom $\{q_0, 0, n-1, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{b} \{q_0, 0, 2, s_3, \ldots, s_t\}$. Množinu vľavo sme vyrobili v bode (5).

(7) Nech $i = 0$ a $s_2 = 2$, $s_3 = 3$. Potom $\{q_0, 0, n-2, 1, s_4 - 2, \ldots, s_t - 2\} \xrightarrow{a}$ $\{q_1, 1, n-1, 2, s_4 - 1, \ldots, s_t - 1\} \xrightarrow{b} \{q_0, 0, 2, 3, s_4, \ldots, s_t\}$. Množinu celkom vľavo sme vyrobili v bode (3).

(8) Nech $i = 1$ ($s_1 = 1$). Potom $\{q_0, 0, s_2 - 1, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_1, 1, s_2, s_3, \ldots, s_t\}$. Množinu vľavo sme vyrobili v bodoch (3)-(6). Tým sme ukázali dosiahnuteľnosť.

Ešte ukážeme rozlíšiteľnosť stavov v podmnožinovom automate M. Treba si uvedomiť, že len jediný stav je zamietajúci a to $\{q_1, 1\}$, všetky ostatné stavy sú akceptujúce.

Zoberme teraz dva rôzne stavy $p = \{q_i\} \cup S$ a $q = \{q_j\} \cup T$ z $\mathcal{R}$. Všimnime si však, že pre každé t, $0 \leq t \leq n-1$, slovo $a^{n-1-t}b(ab^{n-2})^{n-3}$ je akceptované v NFA N len zo stavu t. Z toho vyplýva, že stavy p a q sú rozlíšiteľné, ak $S \neq T$. Uvažujme teraz prípad keď $S = T$. Máme teda $\{q_i\} \cup S$, $\{q_j\} \cup S$, kde $S \subseteq \{0, 1, \ldots, n-1\}$. Nutne $i \neq j$. Najprv si dokážme nasledujúce pomocné tvrdenie.

Lemma 2.3 *Majme dva rôzne prvky r a j z množiny {2,3,…,n-1}. Potom*

$$r \xrightarrow{b^{n-1-r}ab^{r-2}} 0; \quad j \xrightarrow{b^{n-1-r}ab^{r-2}} j$$

Dôkaz.
Prvok r sa posunie do 0, lebo $r \xrightarrow{b^{n-1-r}} n-1 \xrightarrow{a} 0 \xrightarrow{b^{r-2}} 0$. Prvok j je po prvom kole béčok určite menší ako $n-1$ a väčší ako 2. Preto určite bude po jednom áčku rôzny od 0 a 1. Zostane teda v béčkovom cykle. Po ďalších béčkach sme j spolu posunuli o $n-1-r+1+r-2$, čo je $n-2$, čo je veľkosť cyklu na b, a teda sa j vráti samé do seba.

□

Toto tvrdenie ďalej využijeme v dôkaze neekvivalencie, ak $i \neq j$. Rozlíšime niekoľko prípadov:

1) Nech $i \geq 2$, $j = 0$ a $S = \{0, i\} \cup S_1$. Potom

$$\{q_0, 0, i\} \cup S_1 \xrightarrow{\text{Pozorov. 2.3}} \{q_0, 0, i\} \xrightarrow{b^{n-1-i}} \{q_0, 0, n-1\} \xrightarrow{a} \{q_1, 0, 1\} \xrightarrow{a} \{q_2, 0, 1, 2\},$$

$$\{q_i, 0, i\} \cup S_1 \xrightarrow{\text{Pozorov. 2.3}} \{q_i, 0, i\} \xrightarrow{b^{n-1-i}} \{q_{n-1}, 0, n-1\} \xrightarrow{a} \{q_0, 0, 1\} \xrightarrow{a} \{q_1, 1, 2\}.$$

Výsledné množiny sa však líšia v 0, a teda sú rozlíšiteľné.

2) Ostatné prípady prevedieme, aby sme dostali situáciu ako v bode 1).

2a) Nech $i = 0$, $j = 1$ a keďže oba stavy sú z $\mathcal{R}$, tak $S = \{0, 1\} \cup S_1$. Potom $\{q_0, 0, 1\} \cup S_1 \xrightarrow{a} \{q_1, 1, 2\} \cup S_2 \xrightarrow{b} \{q_0, 0, 3\} \cup S_3$ a $\{q_1, 0, 1\} \cup S_1 \xrightarrow{a} \{q_2, 0, 1, 2\} \cup S_2 \xrightarrow{b} \{q_3, 0, 3\} \cup S_3$.

2b) Nech $j = 1$, $i \geq 2$ a keďže oba stavy sú z $\mathcal{R}$, tak $S = \{0, 1, i\} \cup S_1$. Potom $\{q_1, 0, 1, i\} \cup S_1 \xrightarrow{b} \{q_0, 0, i+1\} \cup S_2$ a $\{q_i, 0, 1, i\} \cup S_1 \xrightarrow{b} \{q_{i+1}, 0, i+1\} \cup S_2$.

2c) Nech $i \geq 2$, $j \geq 2$,. Keďže oba stavy sú z $\mathcal{R}$, tak $S = \{0, i, j\} \cup S_1$.

$\{q_i, 0, i, j\} \cup S_1 \xrightarrow{a^{n-i}} \{q_0, 0, n-i, j+n-i\} \cup S_2$ a $\{q_j, 0, i, j\} \cup S_1 \xrightarrow{a^{n-i}} \{q_{j+n-i}, 0, n-i, j+n-i\} \cup S_2$.

□

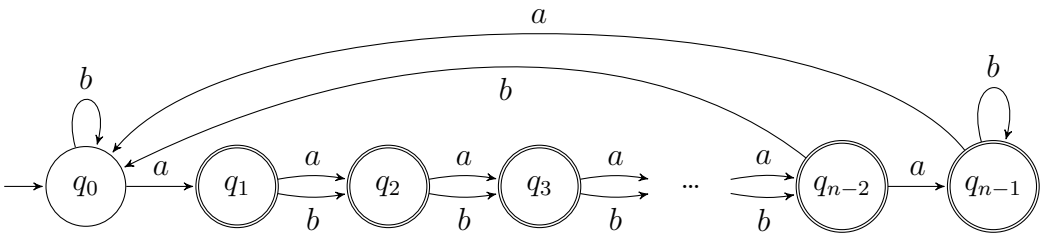
Ukázali sme, že keď DFA pre L má $n-1$ koncových stavov a $q_0 \in F$, tak $\mathrm{sc}(L^2) = (n+2)2^{n-2}$. V ďalšej časti ukážeme, že tento odhad pre štvorec vieme prekonať, ak jediným nekoncovým stavom v DFA pre L zostane počiatočný stav q_0.

2.1.2 Zložitosť štvorca ak $|F| = n-1$ a $q_0 \notin F$

V tejto časti rozoberieme stavovú zložitosť štvorca, ak pôvodný jazyk bol akceptovaný DFA s jediným nekoncovým stavom q_0. V nasledujúcej leme ukážeme horný odhad stavovej zložitosti štvorca v tomto prípade. Jeho tesnosť ukážeme však až na ternárnej abecede. Na binárnej abecede dosiahneme iba o jedna menšiu hodnotu pre zložitosť štvorca.

Lemma 2.4 *Nech $n \geq 3$ a nech L je regulárny jazyk nad abecedou Σ taký, že $\mathrm{sc}(L) = n$ a minimálny DFA pre L má $n-1$ koncových stavov, kde $q_0 \notin F$. Potom $\mathrm{sc}(L^2) \leq (n+3) \cdot 2^{n-2}$ a táto hranica je tesná, ak $|\Sigma| \geq 3$. Hranicu $(n+3) \cdot 2^{n-2} - 1$ vieme dosiahnuť na binárnej abecede.*

Dôkaz.
Najskôr ukážeme horný odhad zložitosti. Všimnime si, že ak v podmnožinovom automate pre L^2 máme dva rôzne stavy $\{q_i\} \cup S$ a $\{q_j\} \cup S$, zjavne $i \neq j$ také, že $\{i, j\} \subseteq S$, tak tieto stavy sú ekvivalentné: Ak w je zamietnuté z p, tak potom pre všetky prvky $s \in S$ platí, že $s \xrightarrow{w} 0$, teda aj pre i, a j. To znamená, že $\{q_j\} \cup S \xrightarrow{w} \{q_0, 0\}$, z čoho vyplýva, že aj stav q zamieta slovo w. Analogicky, ak w je zamietnuté z q, tak je zamietnuté aj z p. Z toho vyplýva, že p a q sú ekvivalentné. Spočítajme teraz množiny, ktoré zostanú neekvivalentné: $\{q_0\} \cup X$, kde $X \subseteq \{0, 1, \ldots, n-1\}$ a $\{q_i, 0\} \cup Y$,



Obr. 2.5: DFA B.

kde $i = 1, 2, \ldots, n-1$, a $Y \subseteq \{1, 2, \ldots, n-1\} \setminus \{i\}$. Takýchto množín je $2^n + (n-1) \cdot 2^{n-2} = (n+3) \cdot 2^{n-2}$. Priestor týchto množín budeme nazývať $\mathcal{R}$.

Zoberme DFA B podľa obrázka 2.5, ktorý bude akceptovať jazyk L. Zostrojme NFA N pre jazyk L^2. Matematickou indukciou podľa veľkosti množín ukážeme, že v M sú všetky množiny z $\mathcal{R}$ dosiahnuteľné. Jedno- a dvojprvkové množiny sú dosiahnuteľné lebo v podmnožinovom automate máme

$$\to \{q_0\} \xrightarrow{a} \{q_1, 0\} \xrightarrow{b} \{q_2, 0\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_{n-2}, 0\} \xrightarrow{b} \{q_0, 0\};$$

$$\{q_{n-2}, 0\} \xrightarrow{a} \{q_{n-1}, 0, 1\} \xrightarrow{b} \{q_{n-1}, 0, 2\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_{n-1}, 0, n-2\} \xrightarrow{b} \{q_{n-1}\},$$

$$\{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\} \xrightarrow{b} \{q_0, 2\} \xrightarrow{b} \cdots \xrightarrow{b} \{q_0, n-2\}.$$

Všimnime si, že sme nevyrobili množinu $\{q_0, n-1\}$. Taká množina nám ani nebude chýbať pri vyrábaní ďalších. Napríklad hneď v bode (1) potrebujeme nanajvýš množinu $\{q_0, n-2\}$, ktorú sme však už dosiahli.

Predpokladajme, že všetky t prvkové množiny z $\mathcal{R}$ sú dosiahnuteľné. Ukážeme, že potom sú všetky $t+1$ prvkové množiny z $\mathcal{R}$ sú dosiahnuteľné. Nech $S = \{q_i, s_1, s_2, \ldots, s_t\}$, kde $q_i \in Q$, a $0 \leq s_1 < s_2 < \cdots < s_t \leq n-1$ je $t+1$ prvková množina z $\mathcal{R}$. Rozlíšime niekoľko prípadov.

(1) Nech $i = 1$, $s_2 \geq 2$. Potom $\{q_0, s_2 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_1, 0, s_2, \ldots, s_t\}$. Množina vľavo je t prvková, teda je dosiahnuteľná podľa indukčného predpokladu.

(2) Nech $1 \leq i \leq n-2$. Indukciou podľa i ukážeme, že každá $t+1$ prvková množina $\{q_i, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$ je dosiahnuteľná. Prvý krok indukcie, $i = 1$, sme ukázali v bode (1). Predpokladajme, že každá $t+1$ prvková množina $\{q_i, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$, kde $1 \leq i \leq n-3$, je dosiahnuteľná. Ukážeme, že potom aj každá $t+1$ prvková množina $\{q_{i+1}, 0, s_2, \ldots, s_t\}$ z $\mathcal{R}$ je dosiahnuteľná. Nech $S = \{q_{i+1}, 0, s_2, \ldots, s_t\}$. Rozlíšime 3 prípady:

(2a) Nech $s_2 = 1$. Potom $\{q_i, 0, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_{i+1}, 0, 1, s_3, \ldots, s_t\} = S$. Všimnime si, že množina vľavo je len t prvková, no je dosiahnuteľná podľa indukčného predpokladu z indukcie podľa t.

(2b) Nech $s_2 \geq 2$ a $s_t \leq n-2$. Potom $\{q_i, 0, s_2 - 1, \ldots, s_t - 1\} \xrightarrow{b} \{q_{i+1}, 0, s_2, s_3, \ldots, s_t\} = S$. Množina vľavo je dosiahnuteľná podľa indukčného predpokladu z indukcie podľa i.

(2c) Nech $s_2 \geq 2$ a $s_t = n-1$. Potom $\{q_i, 0, s_2 - 1, \ldots, s_{t-1} - 1, n-1\} \xrightarrow{b} \{q_{i+1}, 0, s_2, s_3, \ldots, s_{t-1}, n-1\} = S$. Množina vľavo je dosiahnuteľná podľa indukčného predpokladu z indukcie podľa i.

(3) Nech $i = n-1$. Rozlíšime 3 prípady:

(3a) Nech $s_2 = 1$. Potom dostávame $\{q_{n-2}, 0, s_3 - 1, \ldots, s_t - 1\} \xrightarrow{a} \{q_{n-1}, 0, 1, s_3, \ldots, s_t\} = S$. Množina vľavo je dosiahnuteľná podľa bodu (2).

(3b) Nech $s_2 \geq 2$ a $s_t \leq n-2$. Potom $\{q_{n-1}, 0, 1, s_3-s_2+1, \ldots, s_t-s_2+1\}$ $\xrightarrow{b^{s_2-1}}$ $\{q_{n-1}, 0, s_2, s_3, \ldots, s_t\} = S$. Množinu vľavo sme dosiahli v bode (3a).

(3c) Nech $s_2 \geq 1$ a $s_t = n-1$. Potom $\{q_{n-1}, 0, 1, s_3-s_2+1, \ldots, s_{t-1}-s_2+1, n-1\}$ $\xrightarrow{b^{s_2-1}}$ $\{q_{n-1}, 0, s_2, s_3, \ldots, s_{t-1}, n-1\} = S$. Množina vľavo je dosiahnuteľná podľa bodu (3a).

(4) Nech $i = 0$. Rozlíšime niekoľko prípadov:

(4a) Nech $s_1 = 0$ a $s_t \leq n-2$. Potom $\{q_{n-2}, 0, s_2 - 1, \ldots, s_t - 1\} \xrightarrow{b}$ $\{q_0, 0, s_2, \ldots, s_t\} = S$. Množina vľavo je dosiahnuteľná podľa bodu (2).

(4b) Nech $s_1 = 0$ a $s_t = n-1$. Potom $\{q_{n-2}, 0, s_2-1, \ldots, s_{t-1}-1, n-1\} \xrightarrow{b}$ $\{q_0, 0, s_2, \ldots, s_{t-1}, n-1\} = S$. Množina vľavo je dosiahnuteľná v bode (2).

(4c) Nech $s_1 = 1$. Potom $\{q_0, 0, s_2-1, \ldots, s_t-1\} \xrightarrow{b} \{q_0, 1, s_2, \ldots, s_t\} = S$. Množina vľavo je dosiahnuteľná podľa bodu (3).

(4d) Nech $s_1 \geq 2$ a $s_t \leq n-2$. Potom $\{q_0, 1, s_2-s_1+1, \ldots, s_t-s_1+1\} \xrightarrow{b^{s_1-1}}$ $\{q_0, s_1, s_2, \ldots, s_t\} = S$. Množina vľavo je dosiahnuteľná podľa bodu (4c).

(4e) Nech $s_1 \geq 2$ a $s_t = n-1$. Potom $\{q_0, 1, s_2-s_1+1, \ldots, s_{t-1}-s_1+1, n-1\}$ $\xrightarrow{b^{s_1-1}}$ $\{q_0, s_1, s_2, \ldots, s_{t-1}, n-1\} = S$. Množina vľavo sme dosiahli v bode (4c).

Tým sme ukázali dosiahnuteľnosť. Teraz ukážeme, že všetky dosiahnuteľné stavy v M sú navzájom neekvivalentné. Zoberme teraz dva rôzne stavy $p = \{q_i\} \cup S$ a $q = \{q_j\} \cup T$. Všimnime si, že slovo b^n je akceptované v NFA N len zo stavu $n-1$. Slovo $a^{n-1-t}b^n$ je akceptované len zo stavu $0 \leq t \leq n-1$. Z toho vyplýva, že p a q sú rozlíšiteľné, keď $S \neq T$.

Rozoberme teraz situáciu, keď $S = T$. Vtedy nutne $i \neq j$. Predpokladajme, že $0 \leq i < j \leq n-1$ a $\{i, j\} \not\subseteq S$. Nech $i = 0$ a $S \subseteq \{0, 1, \ldots, n-1\}$. Potom $j \notin S$ a máme

$$\{q_0\} \cup S \xrightarrow{a^{n-1-j}} \{q_{n-1-j}\} \cup S' \xrightarrow{b^n} \{q_0, 0\} \xrightarrow{a} \{q_1, 0, 1\};$$

$$\{q_j\} \cup S \xrightarrow{a^{n-1-j}} \{q_{n-1}\} \cup S' \xrightarrow{b^n} \{q_{n-1}, 0\} \xrightarrow{a} \{q_0, 1\}.$$

Výsledné množiny sa líšia v 0. Ak $i \geq 1$, použijeme slovo a^{n-j}, aby sme sa dostali do prípadu vyššie.

Zatiaľ sme dosiahli všetky množiny z $\mathcal{R}$ okrem $\{q_0, n-1\}$. Odtiaľ dostávame hornú hranicu v binárnom prípade, a to $(n+3) \cdot 2^{n-2} - 1$. Na dosiahnutie stavu $\{q_0, n-1\}$ pridáme ďalšie písmeno do B a dodefinujeme prechody na c nasledovne: $\delta(q_0, c) = q_0$; $\delta(q_i, c) = q_{i+1}$, ak $1 \leq i \leq n-2$, a nakoniec $\delta(q_{n-1}, c) = q_0$. V takto doplnenom automate vieme dosiahnuť $\{q_0, n-1\}$ zo stavu $\{q_0, n-2\}$ na c. Dostávame tak tesnú hornú hranicu $(n+3) \cdot 2^{n-2}$.

□

Výsledky lemy 2.2 a 2.4 zhrnieme v nasledujúcom tvrdení.

Veta 2.5 *Nech $n \geq 3$ a nech L je regulárny jazyk nad abecedou Σ taký, že $\mathrm{sc}(L) = n$ a minimálny DFA pre jazyk L má $n-1$ koncových stavov. Potom $\mathrm{sc}(L^2) \leq (n+3) \cdot 2^{n-2}$ a táto hranica je tesná, ak $|\Sigma| \geq 3$. Hranicu $(n+3) \cdot 2^{n-2} - 1$ vieme dosiahnuť na binárnej abecede.*

V binárnom prípade sme pre každé n našli n stavový DFA s jediným nekoncovým stavom taký, že minimálny deterministický automat pre jeho štvorec má $(n+3) \cdot 2^{n-2} - 1$ stavov. Z našich predošlých výpočtov vieme, že táto hranica sa na binárnej abecede nepresiahne ak $n \leq 5$. Domnievame, že pre väčšie hodnoty n to tiež platí, ale zatiaľ to nevieme dokázať.

Kapitola 3

Booleovské a alternujúce automaty

V tejto kapitole zhrnieme známe výsledky týkajúce sa hlavne alternujúcich automatov a predstavíme tak motiváciu hľadania jazyka ťažkého pre štvorec.

Najskôr sa venujeme úvodu do problematiky, kde rozoberieme prechod medzi booleovkými/alternujúcimi a (ne)deterministickými automatmi, a s tým súvisiacu závislosť stavovej zložitosti. Ďalej sa budeme venovať zreťazeniu a predstavíme otvorený problém z roku 1990 ohľadom tesnosti horného odhadu $2^m + n + 1$ pre túto operáciu na alternujúcich automatoch podľa [3]. Následne ukážeme riešenie tohto problému pre štvorec, teda ukážeme tesnosť ohraničenia 2^n+n+1. Tento dôkaz aj zovšeobecníme pre zreťazenie. Nakoniec ukážeme, že zložitosť štvorca na booleovských automatoch je $2^n + n$.

Začnime teda s definíciami základných pojmov. Ich detailnejší popis možno nájsť v [1, 3, 5, 7, 8].

3.1 Definícia

Nech $n \geq 1$. Booleovský automat A definujeme ako päticu $A = (Q, \Sigma, \delta, g_s, F)$, kde Q je množina stavov $q_1, \ldots q_n$, Σ je vstupná abeceda, $F \subseteq Q$ sú koncové stavy. Označme $\mathcal{B}_n$ ako množinu všetkých booleovských funkcií n booleovských premenných $q_1, \ldots, q_n$. Potom ľahko zapíšeme, že $g_s \in \mathcal{B}_n$ je počiatočná funkcia a prechodovú funkciu δ môžme definovať ako zobrazenie do booleovských funkcií, $\delta : Q \times \Sigma \to \mathcal{B}_n$. Prechodovú funkciu teraz rozšírime na $\mathcal{B}_n \times \Sigma^*$ indukciou vzhľadom na dĺžku slova. Nech $g \in \mathcal{B}_n$, $a \in \Sigma$, $w \in \Sigma^*$:

(1) $\delta(g, \varepsilon) = g$;

(2) ak $g = g(q_1, q_2, \ldots, q_n)$ tak $\delta(g, a) = g(\delta(q_1, a), \delta(q_2, a), \ldots, \delta(q_n, a))$;

(3) $\delta(g, aw) = \delta(\delta(g, a), w)$.

Budeme potrebovať ešte vektor finality $f = (f_1, f_2, \ldots, f_n) \in \{0,1\}^n$, taký, že $f_i = 1$, ak $q_i \in F$ a $f_i = 0$, ak $q_i \notin F$, ktorý značí, ktoré stavy sú koncové a ktoré nie. Slovo w je z jazyka akceptovaného AFA A, ak po dosadení finality vektora do $\delta(s, w)$ dostaneme 1, formálne $L(A) = \{w \in \Sigma^* \mid \delta(s, w)(f) = 1\}$.

Ak počiatočnou funkciou BFA je len jediný stav, vtedy takýto automat nazývame alternujúci automat(AFA). Tieto pojmy ilustrujeme na nasledujúcom príklade:

Príklad 3.1 Majme alternujúci automat s dvoma stavmi na binárnej abecede, $A = (\{q_1, q_2\}, \{a, b\}, \delta, q_1, \{q_2\})$. Prechodovú funkciu si definujeme tabuľkou. Poďme zistiť, či je slovo $w = abb$ akceptované. Začneme z počia-

δ	a	b
q_1	$q_1 \vee q_2$	q_1
q_2	q_2	$q_1 \wedge \neg q_2$

točného stavu $s = q_1$, teda $\delta(s, w) = \delta(q_1, abb)$. Podľa tabuľky $\delta(q_1, a)$ je $q_1 \vee q_2$, dostávame teda $\delta(q_1, abb) = \delta(q_1 \vee q_2, bb)$. Pokračujme takto ďalej: $\delta(q_1 \vee q_2, bb) = \delta(q_1 \vee (q_1 \wedge \neg q_2), b) = q_1 \vee (q_1 \wedge \neg(q_1 \wedge q_2))$.

Teraz vyhodnotíme výraz dosadením finality vektora $f = (0, 1)$. Za koncové stavy dosadíme 1, za nekoncové 0. Dostávame $q_1 \vee (q_1 \wedge \neg(q_1 \wedge q_2)) = 0 \vee (0 \wedge \neg(0 \wedge 1)) = 0 \vee (0 \wedge 1) = 0 \vee 0 = 0$. Výledok je 0, teda slovo $w = abb$ náš alternujúci automat A zamieta.

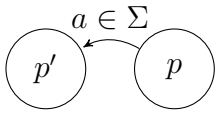
Hovoríme, že AFA A je *minimálny*, ak každý AFA pre jazyk $L(A)$ má aspoň toľko stavov ako A. *Alternujúca stavová zložitosť jazyka* L, $\mathrm{asc}(L)$, je počet stavov v minimálnom AFA pre L. *Booleovská stavová zložitosť jazyka* L, $\mathrm{bsc}(L)$, je počet stavov v minimálnom booleovskom automate pre L.

V nasledujúcej vete ukážeme konštrukciu, ktorá nám dovolí prejsť od booleovského automatu k nedeterministickému s viacerými počiatočnými stavmi (NNFA).

Veta 3.2 ([5, Lema 1]) *Ak L je akceptovaný n stavovým BFA (AFA), tak L je akceptovaný 2^n stavovým NNFA N (s 2^{n-1} počiatočnými stavmi).*

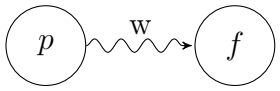
Dôkaz.
Máme daný n stavový BFA $A = (Q, \Sigma, \delta, g_s, F)$ so svojím vektorom finality $f = (f_1, f_2, \ldots, f_n) \in \{0,1\}^n$, kde $Q = \{q_1, q_2, \ldots, q_n\}$. K nemu zostrojíme 2^n stavový NNFA $N = (\{0,1\}^n, \Sigma, \delta', I, \{f\})$ a ukážeme, že $L(A) = L(N)$.

Za stavy v N zoberieme všetky reťazce núl a jednotiek dĺžky n, je ich presne 2^n. Množinu počiatočných stavov budú tvoriť tie reťazce, ktoré vyhovujú počiatočnej funkcii g_s v A, teda $I = \{p \in \{0,1\}^n \mid g_s(p) = 1\}$. Koncový stav bude len jeden, a to binárny vektor f. Prechodovú funkciu



$\delta' : \{0,1\}^n \times \Sigma \to \{0,1\}^n$ definujeme ako $\delta'(p,a) = \{p' \in \{0,1\}^n \mid \big(\delta(q_1,a), \delta(q_2,a), \ldots, \delta(q_n,a)\big)(p') = p\}$. Čo znamená, že ak chceme zistiť, aké šípky vchádzajú do stavu $p' \in \{0,1\}^n$ na písmenko a, tak vyhodnotíme $\delta(q,a)(p')$, pre každé $q \in Q$. Dostaneme tak nový binárny vektor $p \in \{0,1\}^n$, pre ktorý bude platiť $\delta'(p,a) \to p'$.

Ukážeme najskôr pomocné tvrdenie, ktoré hovorí, že keď sa vieme v N dostať zo stavu p do f na slovo w, tak je to ekvivalenté s tým, že v A po dosadení finality vektora f na slovo w dostaneme p. Teda



$$\Leftrightarrow \quad \big(\delta(q_1,w), \delta(q_2,w), \ldots, \delta(q_n,w)\big)(f) = p.$$

Lemma 3.3 *V NNFA N existuje výpočet prevádzajúci stav p na stav f na slovo w práve vtedy, keď v BFA A platí $\big(\delta(q_1,w), \delta(q_2,w), \ldots, \delta(q_n,w)\big)(f) = p$.*

Dôkaz.
Tvrdenie ukážeme indukciou podľa dĺžky slova $w \in \Sigma^*$.

1. Nech $|w| = 0$, teda $w = \varepsilon$. Z toho vyplýva, že $p \xrightarrow{\varepsilon} f$. To znamená, že $p = f$. Potom $\big(\delta(q_1,\varepsilon), \ldots, \delta(q_n,\varepsilon)\big)(f) = (q_1, \ldots, q_n)(f) = f = p$.

2. Indukčný predpoklad: Ak $|w| = l$, tak platí naše tvrdenie, že

$$p \xrightarrow{w} f \quad \Leftrightarrow \quad \big(\delta(q_1,w), \delta(q_2,w), \ldots, \delta(q_n,w)\big)(f) = p.$$

Ukážeme, že potom tvrdenie platí pre slová dĺžky $l+1$. Zoberme slovo w dĺžky $l+1$, ktoré vieme rozdeliť na $w = a \cdot v$, kde $a \in \Sigma$ a v je slovo dĺžky l.

$\Leftarrow$
Zoberme $p = \big(\delta(q_1,w), \ldots, \delta(q_n,w)\big)(f)$. Ukážeme, že existuje cesta $p \rightsquigarrow f$ na slovo w.

$$p = \big(\delta(q_1,w), \ldots, \delta(q_n,w)\big)(f) = \big(\delta(q_1,a), \ldots, \delta(q_n,a)\big) \underbrace{\big(\delta(q_1,v), \ldots, \delta(q_n,v)\big)(f)}_{p'}$$

Na $\big(\delta(q_1,v), \ldots, \delta(q_n,v)\big)(f)$ sa vzťahuje indukčný predpoklad, lebo $|v| = l$, a teda existuje cesta $p' \rightsquigarrow f$ na slovo v, čiže $p = \big(\delta(q_1,a), \ldots, \delta(q_n,a)\big)(p')$. Potom podľa našej definície δ' v N, môžeme písať, že existuje $p \xrightarrow{a} p' \xrightarrow{v} f$, z čoho vyplýva, že $p \xrightarrow{w} f$.

$\Rightarrow$
Vieme, že existuje cesta $p \xrightarrow{w} f$. Tú vieme rozdeliť na $p \xrightarrow{a} p' \xrightarrow{v} f$. Keďže

$|v| = l$ využijeme indukčný predpoklad $\big(\delta(q_1, v), \ldots, \delta(q_n, v)\big)(f) = p'$. Keďže $p \xrightarrow{a} p'$, tak platí aj $p = \big(\delta(q_1, a), \ldots, \delta(q_n, a)\big)(p')$, z toho

$$p = (\delta(q_1, a), \ldots, \delta(q_n, a))(\delta(q_1, v), \ldots, \delta(q_n, v))(f) = (\delta(q_1, w), \ldots, \delta(q_n, w))(f).$$

□

Teraz ešte ukážme, že naozaj platí, že slovo w je akceptované pomocou BFA A, práve vtedy, keď je akceptované v NNFA N.

Lemma 3.4 *Nech $L(A)$ je jazyk akceptovaný BFA A a $L(N)$ je jazyk akceptovaný NNFA N. Potom $L(A) = L(N)$.*

Dôkaz.
⇐
Nech $w \in L(N)$. Existuje teda výpočet z počiatočného stavu p taký, že $\to p \xrightarrow{w} f$. Či BFA A akceptuje slovo w rozhodne výsledok $\delta(g_s, w)(f)$.

$$\delta(g_s, w)(f) = g_s\big(\underbrace{\big(\delta(q_1, w), \ldots, \delta(q_n, w)\big)(f)}_{p}\big) = g_s(p)$$

Keďže p je počiatočný stav a pre počiatočné stavy v N platí, že vyhovujú počiatočnej funkcii g_s v BFA A, tak výsledok je $\delta(g_s, w)(f) = g_s(p) = 1$. Pre nás to ale znamená, že ak je slovo akceptované v NNFA N, tak je akceptované aj v BFA A.

⇒
Nech $w \in L(A)$. Platí teda $1 = \delta(g_s, w)(f) = g_s\big(\underbrace{\big(\delta(q_1, w), \ldots, \delta(q_n, w)\big)(f)}_{p}\big)$.
V N teda existuje cesta $p \xrightarrow{w} f$ do jediného koncového stavu v N. K tomu ešte platí $g_s(p) = 1$, z čoho je jasné, že p musí byť počiatočným stavom v N. Slovo w teda tiež padne do jazyka $L(N)$. Dostávame, že ak je slovo akceptované v BFA A, tak je akceptované aj v NNFA N.

□

Došli sme teda k tomu, že ak máme n stavový BFA, vieme k nemu zostrojiť 2^n stavový NNFA, ktorý akceptuje rovnaký jazyk. Všimnime si, že ak daný BFA bol alternujúci, tak potom mal jediný počiatočný stav q_i. Z toho vyplýva, že konštruovaný NNFA bude mať presne 2^{n-1} počiatočných stavov, lebo počiatočnými stavmi budú tie booleovské vektory, ktorých i-ta zložka je 1.

□

Túto transformáciu z booleovského automatu na NNFA teraz ukážeme na príklade.

Príklad 3.5 Majme BFA $A = \{Q, \Sigma, \delta, g_s, F = \{q_3\}\}$ na binárnej abecede definovaný nasledovnou tabuľkou.

δ	a	b
$\rightarrow q_1$	q_3	q_1
q_2	$q_1 \vee q_2$	q_3
q_3	$\neg q_1$	$q_1 \wedge \neg q_2$

Vidíme, že $g_s = q_1$, vektor finality je $f = (0, 0, 1)$. Zostrojme k A NNFA automat $N = \{Q', \Sigma, \delta', I, F'\}$, ktorý bude tiež rozoznávať jazyk $L(A)$. Využijeme predchádzajúcu vetu.

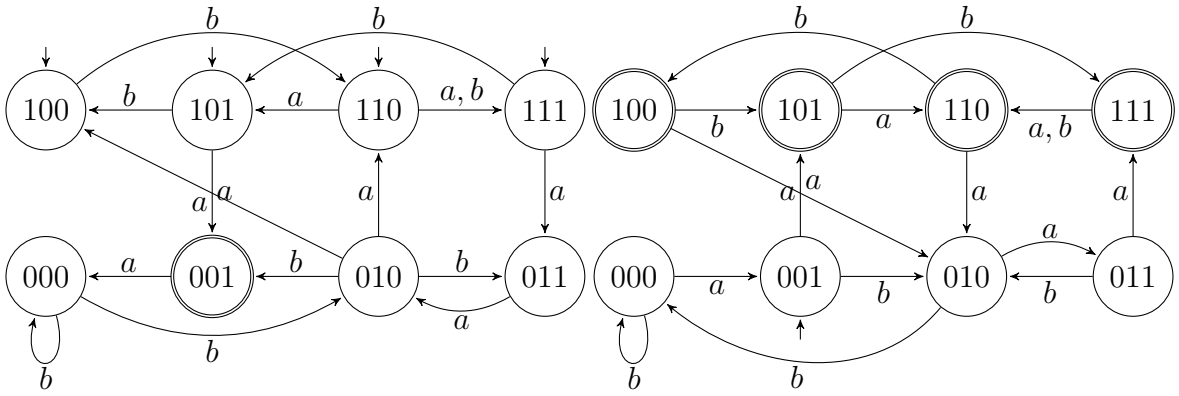
Stavy budú binárne reťazce dĺžky 3, $Q' = \{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Množina počiatočných stavov bude $I = \{p \in \{0,1\} \mid g_s(p) = 1\} = \{100, 101, 110, 111\}$. Množinu koncových stavov bude tvoriť jediný reťazec $F = \{f\} = \{001\}$. Určíme ešte prechodovú funkciu δ'.

$\big(\delta(q_1, a), \delta(q_2, a), \delta(q_3, a)\big)(000) = \big(q_3(000), q_1 \vee q_2(000), \neg q_1(000)\big) = (0, 0, 1)$
$\ldots \quad \delta'(001, a) \ni (000)$
$\big(\delta(q_1, b), \delta(q_2, b), \delta(q_3, b)\big)(000) = \big(q_1(000), q_3(000), q_1 \wedge \neg q_2(000)\big) = (0, 0, 0)$
$\ldots \quad \delta'(000, b) \ni (000)$

Keď počítame $\delta'(p, a)$, tak by sme mali zobrať postupne všetky reťazce p' z $\{0,1\}^3$ a do množiny $\delta'(p, a)$ by padli len tie, spĺňajúce podmienku $\big(\delta(q_1, a), \delta(q_2, a), \delta(q_3, a)\big)(p') = p$. My si však urýchlime výpočet a pýtame sa, čo vchádza (aký stav p) na písmeno a do stavu p'.

Navyše $\delta(q_1, a), \delta(q_2, a), \delta(q_3, a)$ nie je nič iné, ako prvý stĺpec prechodovej funkcie v booleovskom automate A. Dopočítajme δ'.

$\big(\delta(q_1, a), \delta(q_2, a), \delta(q_3, a)\big)(001) = \big(q_3(001), q_1 \vee q_2(001), \neg q_1(001)\big) = (1, 0, 1)$
$\ldots \quad \delta'(101, a) \ni (001)$
$\big(\delta(q_1, b), \delta(q_2, b), \delta(q_3, b)\big)(001) = \big(q_1(001), q_3(001), q_1 \wedge \neg q_2(001)\big) = (0, 1, 0)$
$\ldots \quad \delta'(010, b) \ni (001)$
$\big(\delta(q_1, a), \delta(q_2, a), \delta(q_3, a)\big)(010) = \big(q_3(010), q_1 \vee q_2(010), \neg q_1(010)\big) = (0, 1, 1)$
$\ldots \quad \delta'(011, a) \ni (010)$
$\big(\delta(q_1, b), \delta(q_2, b), \delta(q_3, b)\big)(010) = \big(q_1(010), q_3(010), q_1 \wedge \neg q_2(010)\big) = (0, 0, 0)$
$\ldots \quad \delta'(000, b) \ni (010)$
$\big(\delta(q_1, a), \delta(q_2, a), \delta(q_3, a)\big)(011) = \big(q_3(011), q_1 \vee q_2(011), \neg q_1(011)\big) = (1, 1, 1)$
$\ldots \quad \delta'(111, a) \ni (011)$
$\big(\delta(q_1, b), \delta(q_2, b), \delta(q_3, b)\big)(011) = \big(q_1(011), q_3(011), q_1 \wedge \neg q_2(011)\big) = (0, 1, 0)$
$\ldots \quad \delta'(010, b) \ni (011)$



Obr. 3.6: NNFA N (vľavo) a N^R (vpravo) z príkladu 3.5.

$\big(\delta(q_1,a),\delta(q_2,a),\delta(q_3,a)\big)(100) = \big(q_3(100), q_1 \vee q_2(100), \neg q_1(100)\big) = (0,1,0)$
$\ldots \quad \delta'(010,a) \ni (100)$
$\big(\delta(q_1,b),\delta(q_2,b),\delta(q_3,b)\big)(100) = \big(q_1(100), q_3(100), q_1 \wedge \neg q_2(100)\big) = (1,0,1)$
$\ldots \quad \delta'(101,b) \ni (100)$
$\big(\delta(q_1,a),\delta(q_2,a),\delta(q_3,a)\big)(101) = \big(q_3(101), q_1 \vee q_2(101), \neg q_1(101)\big) = (1,1,0)$
$\ldots \quad \delta'(110,a) \ni (101)$
$\big(\delta(q_1,b),\delta(q_2,b),\delta(q_3,b)\big)(101) = \big(q_1(101), q_3(101), q_1 \wedge \neg q_2(101)\big) = (1,1,1)$
$\ldots \quad \delta'(111,b) \ni (101)$
$\big(\delta(q_1,a),\delta(q_2,a),\delta(q_3,a)\big)(110) = \big(q_3(110), q_1 \vee q_2(110), \neg q_1(110)\big) = (0,1,0)$
$\ldots \quad \delta'(010,a) \ni (110)$
$\big(\delta(q_1,b),\delta(q_2,b),\delta(q_3,b)\big)(110) = \big(q_1(110), q_3(110), q_1 \wedge \neg q_2(110)\big) = (1,0,0)$
$\ldots \quad \delta'(100,b) \ni (110)$
$\big(\delta(q_1,a),\delta(q_2,a),\delta(q_3,a)\big)(111) = \big(q_3(111), q_1 \vee q_2(111), \neg q_1(111)\big) = (1,1,0)$
$\ldots \quad \delta'(110,a) \ni (111)$
$\big(\delta(q_1,b),\delta(q_2,b),\delta(q_3,b)\big)(111) = \big(q_1(111), q_3(111), q_1 \wedge \neg q_2(111)\big) = (1,1,0)$
$\ldots \quad \delta'(110,b) \ni (111)$

3.1.1 Zrkadlový obraz

Všeobecne vytvoríme automat M^R pre zrkadlový obraz jazyka $L(M)$ akceptovaného automatom M, tak že v automate M otočíme každú jednu šípku, a koncové stavy budú počiatočnými a počiatočné koncovými.

Formálne, ak pôvodné $M = (Q, \Sigma, \delta, I, F)$ tak $M^R = (Q, \Sigma, \delta^R, F, I)$, kde

$\delta^R : Q \times \Sigma \to 2^Q$, $\delta^R(q,a) = \{p \in Q \mid q \in \delta(p,a)\}$. Potom je platí, že $L(M^R) = (L(M))^R$.

Uvedomme si, že zkradlový obraz NNFA N, podľa konštrukcie podľa lemy 3.2, je úplný DFA lebo:

- V NNFA N je len 1 koncový stav, takže N^R má len jeden počiatočný.
- V N platí, že do každého stavu vchádza práve jedna šípka pre každé $a \in \Sigma$. Potom, po otočení automatu v N^R platí, že z každého stavu vychádza práve jedna šípka pre každé $a \in \Sigma$.

Dostávame teda nasledujúci výsledok:

Veta 3.6 *Ak jazyk L je akceptovaný n stavovým BFA (AFA), tak L^R je akceptovaný 2^n stavovým DFA (2^n stavový DFA, v ktorom je 2^{n-1} stavov koncových).*

Dôsledok 3.7 *Nech L je regulárny jazk. Potom* $\mathrm{bsc}(L) \geq \lceil \log(\mathrm{sc}(L^R)) \rceil$ *a* $\mathrm{asc}(L) \geq \lceil \log(\mathrm{sc}(L^R)) \rceil$.

Dôkaz.
Ak by nejaký BFA pre L mal k stavov, kde $k < \log(\mathrm{sc}(L^R))$, tak potom podľa vety 3.6 jazyk L^R by bol akceptovaný DFA s $2^k < \mathrm{sc}(L^R)$ stavmi, čo by bol spor. Dôkaz pre asc by bol rovnaký.

□

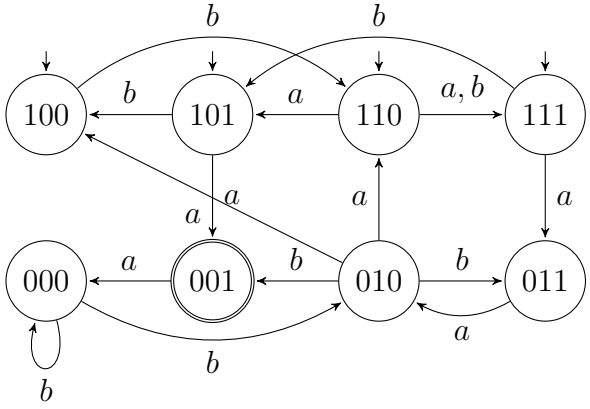
Dôsledok 3.8 *Nech jazyk L je akceptovaný n stavovým AFA. Potom minimálny DFA pre L^R má najviac 2^{n-1} koncových stavov.*

Dôkaz.
Podľa vety 3.6 je L^R akceptovaný 2^n stavovým DFA, v ktorom 2^{n-1} stavov je koncových. Minimalizáciou tohto DFA počet koncových stavov môže iba klesnúť.

□

Už vieme prejsť od n stavového BFA pre L k 2^n stavovému DFA pre L^R podľa vety 3.2. Teraz ukážeme, že je možný aj opačný prechod.

Veta 3.9 ([5, Lema 2]) *Ak L je akceptovaný 2^n stavovým NNFA N a platí, že N^R je DFA. Potom L je akceptovaný n stavovým BFA. Navyše, ak N má 2^{n-1} stavov koncových, tak potom L je akceptovaný n stavovým AFA.*



Obr. 3.7: NNFA N z príkladu 3.5

Dôkaz.
Nech NNFA N vyzerá takto $N = (Q, \Sigma, \delta, I, F = \{f\}, f \in Q)$. NNFA N musí mať práve jeden koncový stav, lebo z predpokladov vety N^R je už DFA. Stavy v NNFA N si očíslujeme. Tým pádom $Q = \{0, 1, 2, \ldots, 2^n - 1\}$ a $F = \{k\}$.

Označme $\mathrm{BIN}(i) \in \{0,1\}^n$ binárny zápis čísla i na n bitov, zľava doplnených nulami. Zrejme $f = \mathrm{BIN}(k) = f_1, f_2, \ldots, f_n$, kde $f_i \in \{0,1\}$. Definujme si teraz BFA $A = (Q', \Sigma, \delta', g_s, F')$ nasledovne:

- Nech $Q' = \{q_1, \ldots, q_n\}$. Máme teda n stavový BFA.
- Množina koncových stavov $F' = \{q_i \mid f_i = 1\}$.
- Počiatočnú funckiu g_s predpíšeme takto: $g_s(\mathrm{BIN}(i)) = 1$ práve vtedy, keď $i \in I$ a podobne $g_s(\mathrm{BIN}(i)) = 0$ práve vtedy, keď $i \notin I$. Vznikne tabuľka pre g_s, z ktorej už vieme vyčítať jej funkčný zápis.
- Prechodovú funkciu $\delta' : Q' \times \Sigma \to \mathcal{B}_n$ takto

  $$\big(\delta'(q_1, a), \delta'(q_2, a), \ldots, \delta'(q_n, a)\big)(\mathrm{BIN}(i)) = \mathrm{BIN}(j) = \mathrm{BIN}(\delta^R(i, a)), \text{ pre } a \in \Sigma,$$

  takého stavu j, že $j \xrightarrow{a} i$ v N, a taký je len jeden, lebo N^R je DFA, keďže δ^R je deterministická funkcia.

Indukciou, podobne ako vo Vete 3.2 ukážeme, že $L(A) = L(N)$.

□

Príklad 3.10 Vytvorme teraz BFA k NNFA N z príkladu 3.5, ktorý máme zadaný na obrázku 3.7.

Máme 8 stavov, $8 = 2^3$, takže naše budúce BFA A, bude mať tri stavy $\{q_1, q_2, q_3\}$. Taktiež NNFA N spĺňa predpoklady vety, to jest, že jeho otočením vznikne DFA. Má len jeden koncový stav, a to 001. Očíslujme si

stavy v N. Napríklad podľa toho, aké decimálne číslo konkrétny binárny vektor predstavuje. Máme teda NNFA $N = (Q, \{a, b\}, \delta, I, F = \{f\})$, kde $Q = \{0, 1, 2, 3, 4, 5, 6, 7\}$ a v tomto prípade je $F = \{001\} = \{1\}$.

Prejdime teraz k definovaniu BFA $A = (Q', \Sigma, \delta', g_s, F')$. Ako sme už povedali, $Q' = \{q_1, q_2, q_3\}$. Vytvorme teraz množinu koncových stavov q_i, kde i-ty bit vo vektore f je jednotka. V našom prípade $F' = \{q_3\}$.

V pôvodnom automate tvorili množinu počiatočných stavov binárne vektory 100,101,110,111. Vytvorme teda tabuľku pravdivostných hodnôt pre g_s.

(q_1, q_2, q_3)	g_s
000	0
001	0
010	0
011	0
100	1
101	1
110	1
111	1

Máme g_s zadanú tabuľkou, potrebujeme nájsť jej funkčný zápis. Spravíme disjunkciu (OR) riadkov, kde je $g_s = 1$. V našom prípade
$g_s = (q_1 \wedge \neg q_2 \wedge \neg q_3) \vee (q_1 \wedge \neg q_2 \wedge q_3) \vee (q_1 \wedge q_2 \wedge \neg q_3) \vee (q_1 \wedge q_2 \wedge q_3)$.

Pre definovanie prechodovej funkcie $\delta' : Q' \times \Sigma \to \mathcal{B}_n$, takisto najskôr vytvoríme tabuľku z ktorej potom budeme vedieť vyčítať funkčný predpis. Z predošlej vety vieme, že prechodovú funkciu zadefinujeme ako

$\big(\delta'(q_1, a), \delta'(q_2, a), \delta'(q_3, a)\big)(\mathrm{BIN}(i)) = \mathrm{BIN}(j) = \mathrm{BIN}(\delta^R(i, a))$ takého stavu j, že $j \xrightarrow{a} i$, kde i, j sú stavy v N.

(q_1, q_2, q_3)	$\delta'(q_1, a)$,	$\delta'(q_2, a)$	$\delta'(q_3, a)$	$\delta'(q_1, b)$	$\delta'(q_2, b)$	$\delta'(q_3, b)$
000	0	0	1	0	0	0
001	1	0	1	0	1	0
010	0	1	1	0	0	0
011	1	1	1	0	1	0
100	0	1	0	1	0	1
101	1	1	0	1	1	1
110	0	1	0	1	0	0
111	1	0	0	1	1	0

Zoberme si napríklad prvý riadok. Ten odpovedá zápisu

$$\big(\delta'(q_1, a), \delta'(q_2, a), \delta'(q_3, a)\big)(000) = (001).$$

V pôvodnom automate N to odpovedá prechodu $000 \xleftarrow{a} 001$. Stĺpec $\delta'(q_1, a)$ je tabuľka pravdivostných hodnôt booleovskej funkcie ktorú aplikujeme v BFA A pri čítaní písmena a zo stavu q_1.

Poznáme už všetko v novom BFA $A = (Q', \{a, b\}, \delta', g_s, F')$, takže môžeme zapísať, že $Q' = \{q_1, q_2, q_3\}$, množina koncových stavov F' obsahuje len

stav q_3. Počiatočná funkcia je $g_s = q_1$ a prechodovú funckiu vieme zapísať tabuľkou:

	a	b
q_1	q_3	q_1
q_2	$q_1 \vee q_3$	q_3
q_3	$\neg q_1$	$q_1 \wedge \neg q_2$

3.2 Zreťazenie na alternujúcich automatoch

V tejto stati sa budeme venovať zreťazeniu na alternujúcich automatoch. Najskôr uvedieme lemu o zreťazení jazyka akceptovaného NFA a jazyka akceptovaného AFA. Na základe tohto poznatku budeme vedieť odvodiť horný odhad stavovej zložitosti pre zreťazenie jazykov, ktoré sú akceptované m a n stavovými alternujúcimi automatmi.

Ak $A = (Q, \Sigma, \delta, s, F)$ je AFA, kde $\delta(q, a)$ je konštanta alebo booleovská funkcia iba s $\vee$-operátorom pre každé $q \in Q$ a $a \in \Sigma$, potom sa A správa ako NFA. Hovoríme, že A je *NFA v AFA reprezentácii*.

Lemma 3.11 ([3, Veta 9.2]) *Nech $A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ je NFA v AFA reprezentácii a nech $A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ je ľubovoľný AFA taký, že $Q_1 \cap Q_2 = \emptyset$. Uvažujme AFA $A_1 \cdot A_2 = (Q_1 \cup Q_2, \Sigma, \delta, s_1, F_2)$ taký, že pre všetky stavy $q \in Q_1 \cup Q_2$ a všetky písmená $a \in \Sigma$ je prechodová funkcia takáto:*

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{ak } q \in Q_1 \text{ a } q \notin F_1; \\ \delta_1(q,a) \vee \delta_2(s_2,a), & \text{ak } q \in Q_1 \text{ a } q \in F_1; \\ \delta_2(q,a), & \text{ak } q \in Q_2. \end{cases}$$

Potom $L(A_1 \cdot A_2) = L(A_1) \cdot L(A_2)$.

Dôkaz.
Pripomeňme, že AFA A_1 a A_2 majú svoje príslušné vektory finality f_1 a f_2. AFA $A_1 \cdot A_2$ má svoj vektor finality dĺžky $|f_1| + |f_2|$, kde prvých f_1 prvkov sú nuly a ďalších f_2 prvkov sú presne prvky vektora f_2, keďže koncové stavy AFA $A_1 \cdot A_2$ sú tvorené množinou F_2.

Nech $w \in L(A_1) \cdot L(A_2)$. Potom $w = xy$, kde $x = x_1x_2 \dots x_k \in L(A_1)$ a $y = y_1y_2 \dots y_\ell \in L(A_2)$.

Výpočet na AFA $A_1 \cdot A_2$ pre w bude:
$\delta(s_1, w)(f) = \delta(s_1, x1 \dots x_k y_1 \dots y_\ell)(f) = \delta(\delta(s_1, x1)x_2 \dots x_k y_1 \dots y_\ell)(f)$.

Keďže $s_1 \in Q_1$, podľa predpisu δ sa prvých k písmen, teda $x_1, \dots, x_k$ bude čítať ako v NFA A_1, čo vytvára disjunkciu stavov z Q_1. Ak prejdeme cez nejaký koncový stav z F_1, tak do disjunkcie pribudnú členy $\delta_2(s_2, a)$,

čo môžeme zapísať ako booleovskú funkciu $g = \bigvee_{q \in R} q \vee \bigvee_{q \in R \cap F_1} \delta_2(s_2, x_q)$, kde R je podmnožina stavov z Q_1, ktoré pri čítaní x dosiahneme. Keďže $x \in L(A_1)$, tak v g sa určite bude vyskytovať aspoň jeden koncový stav z F_1.

Po prečítaní slova x dostávame:
$\delta(s_1, w)(f) = \cdots = \delta(g, y)(f) = \delta(\delta_2(s_2, y_1), y_2 \dots y_\ell)(f) = \delta_2(s_2, y)(f_2) = 1$.
Funkcia g je len disjunkciou stavov z Q_1 a iných booleovských funkcií δ_2. Pri vyhodnocovaní vo finality vektore f preto môžeme zabudnúť na nuly, ktoré vzniknú zo stavov z Q_1 a zamietajúcich výpočtov z δ_2. Slovo $y \in L(A_2)$, čo znamená, že $\delta_2(s_2, y)(f_2) = 1$. Slovo $w \in L(A_1)L(A_2)$ náš alternujúci automat $A_1 \cdot A_2$ akceptuje.

Teraz ukážeme opačnú stranu, a to, čo sa deje pre slová, ktoré nepatria do zreťazenia $L(A_1)$ a $L(A_2)$. Nech $w \notin L(A_1)L(A_2)$. Označme $y_1, y_2, \dots, y_n$ také slová, ktoré sú sufixami w a patria do jazyka $L(A_2)$.

Ak $n = 0$, tak w nemá žiaden sufix, ktorý by bol v $L(A_2)$. Vtedy AFA $A_1 \cdot A_2$ zamieta w, lebo výsledná funkcia, ktorá sa vyhodnotí vo vektor f je disjukcia stavov z Q_1, čo pridáva len nuly, a booleovských funkcií δ_2, ktoré tiež prispievajú len nulami, lebo w nemá sufix z $L(A_2)$.

Ak $n > 0$, potom existujú navzájom rôzne slová $x_1, x_2, \dots, x_n \in \Sigma^*$ také, že $x_i y_i = w$ pre $i = 1, 2, \dots, n$. Zjavne $x_i \notin L(A_1)$, lebo potom by $w \in L(A_1)L(A_2)$. Keďže x_i nie je z jazyka $L(A_1)$ pre každý sufix $y_i \in L(A_2)$, tak v $A_1 \cdot A_2$ nenastane situácia, že po dočítaní prvej časti slova bude vo vznikajúcej disjunkcii koncový stav z F_1, ktorý by spôsobil akceptovanie druhej časti slova. Z toho vyplýva, že $w \notin L(A_1 \cdot A_2)$.

□

Pre zreťazenie dvoch alternujúcich automatov A_1 s m stavmi a A_2 s n stavmi využijeme predchádzajúcu lemu, no potrebujeme zreťaziť NFA v AFA reprezentácii. Alternujúci automat A_1 však vieme podľa vety 3.2 prerobiť na 2^m stavový NNFA. Takýto NNFA môže mať nekoľko počiatočných stavov. Tých sa zbavíme pridaním jedného počiatočného stavu, ktorý ďalej napríklad na ε prechody prejde do pôvodných počiatočných stavov. Získali sme $2^m + 1$ stavový NFA A_1', ktorý vieme zapísať do AFA reprezentácie a jazyk ním akceptovaný sa nezmenil $L(A_1') = L(A_1)$. Na zreťazenie jazykov $L(A_1')$ a $L(A_2)$ už môžeme priamo použiť lemu 3.11. Dostávame teda nasledujúce tvrdenie.

Veta 3.12 ([3, Veta 9.3]) *Nech A_1 a A_2 sú AFA s m a n stavmi. Potom existuje AFA A s najviac 2^m+n+1 stavmi, ktorá akceptuje jazyk $L(A_1)L(A_2)$.*

Z tejto vety vyplýva horný odhad stavovej zložitosti zreťazenia. Autori z [3] však tesnosť tohto odhadu nechali otvorenou. V práci [5] z roku 2012 bola dosiahnutá zložitosť $2^m + n$ na binárnej abecede. Tam použité automaty

avšak nedosahujú hornú hranicu pre zreťazenie ak v druhom z nich je viac nekoncových stavov. Takže problém zostáva nevyriešený.

Ďalším priamym dôsledkom tejto vety je horný odhad $2^n + n + 1$ pre zložitosť štvorca na alternujúcich automatoch. Týmto problémom sa budeme zaoberať v nasledujúcej časti.

3.3 Štvorec na alternujúcich automatoch

V [3] je formulovaný problém tesnosti horného ohraničenia $2^m + n + 1$ pre zreťazenie na AFA. Pre štvorec na alternujúcich automatoch z toho vyplýva podobný problém tesnosti horného ohraničenia $2^n + n + 1$. V nasledujúcej vete podávame dôkaz tohto problému pre štvorec. Využijeme náš výsledok o ťažkom jazyku pre štvorec na deterministických automatoch, ktorý sme ukázali v kapitole 2.

Veta 3.13 *Nech L je regulárny jazyk nad abecedou Σ taký, že $\mathrm{asc}(L) = n$, kde $n \geq 2$. Potom $\mathrm{asc}(L^2) \leq n + 2^n + 1$, a táto hranica je tesná, ak $|\Sigma| \geq 2$.*

Dôkaz.
Pre dôkaz tesnosti využijeme automat z Obr. 2.1. Nech L^R je jazyk akceptovaný DFA A z Obr. 2.1 s 2^n stavmi, kde polovica stavov ($k = 2^{n-1}$) je koncových. Podľa vety 3.9, že L je akceptovaný n stavovým AFA. Vytvorme teraz štvorec pre náš jazyk L^R. Ľahko vidno, že $(L \cdot L)^R = L^R \cdot L^R = (L^R)^2$. Podľa dôkazu Lemy 2.1 spočítajme, koľko stavov by potreboval DFA pre takýto jazyk:

$$\mathrm{sc}\big((L^R)^2\big) = (2^n - 2^{n-1}) \cdot 2^{2^n} + (2^{n-1}) \cdot 2^{2^n - 1}.$$

Podľa dôsledku 3.7 z toho vyplýva, že ak prejdeme k alternujúcim automatom, tak

$$\mathrm{asc}(L^2) \geq \lceil \log\big((2^n - 2^{n-1}) \cdot 2^{2^n} + (2^{n-1}) \cdot 2^{2^n - 1}\big) \rceil = n + 2^n.$$

Z [3] však vieme, že $\mathrm{asc}(L^2) \leq 2^n + n + 1$. Predpokladajme sporom, že L^2 je akceptované $2^n + n$ stavovou AFA. Potom $(L^2)^R$ je akceptovaný 2^{n+2^n} stavovým DFA s $1/2 \cdot (2^{n+2^n})$ koncovými stavmi. V minimálnom DFA pre $(L^2)^R$ je teda najviac $1/2 \cdot (2^{n+2^n})$ koncových stavov. Minimálny DFA pre $(L^2)^R$ má $(2^n - 2^{n-1}) \cdot 2^{2^n} + (2^{n-1}) \cdot 2^{2^n - 1} = 2^{n-1} \cdot 2^{2^n} + 2^{n-1} \cdot 2^{2^n - 1}$ stavov. Z toho je $2^{n-1} \cdot 2^{2^{n-1}} + 2^{n-1} \cdot 2^{2^{n-1} - 1}$ nekoncových. Keď odčítame jedno od druhého, tak po úpravách zistíme, že koncových stavov je

$$2^{n-1} \cdot \big(2^{2^n} - 2^{2^{n-1}}\big) + 2^{n-1} \cdot \big(2^{2^n - 1} - 2^{2^{n-1} - 1}\big) = 2^{n-1} \cdot 2^{2^n} \cdot \left(1 + \frac{1}{2} - \frac{1}{2^{2^{n-1}}} - \frac{1}{2^{2^{n-1}+1}}\right).$$

To vieme zdola odhadnúť pre $n \geq 2$ takto:

$$2^{n-1} \cdot 2^{2^n} \cdot \left(1 + \frac{1}{2} - \frac{1}{2^{2^{n-1}}} - \frac{1}{2^{2^{n-1}+1}}\right) > 2^{n-1} \cdot 2^{2^n} \cdot \left(1 + \frac{1}{2} - \frac{1}{4} - \frac{1}{4}\right) = 2^{n+2^n-1}.$$

Teraz nám však vychádza, že máme mať určite viac koncových stavov ako $1/2 \cdot (2^{n+2^n})$, čo je spor s predpokladom, že AFA pre jazyk L^2 má $2^n + n$ stavov. Takže $\text{asc}(L^2) \geq n + 2^n + 1$.

□

Týmto sme ukázali, že horný odhad $2^n + n + 1$ pre zložitosť operácie štvorec na alternujúcich automatoch je tesný na binárnej abecede.

Tento náš výsledok môžeme zovšeobecniť na zreťazenie a to tak, že zoberieme automaty A a B ako na obrázku 2.1 modifikované takto: DFA A bude mať m stavov, z ktorých posledných k, kde $1 \leq k \leq m-2$, je koncových. DFA B bude mať n stavov, z ktorých posledných ℓ, kde $1 \leq \ell \leq n-1$, je koncových.

Potom dôkaz lemy 2.1 pozmeníme tak, že vieme ukázať, že zreťazenie týchto dvoch jazykov dosahuje zložitosť $(m-k) \cdot 2^n + k \cdot 2^{n-1}$. Z toho vyplýva, že podobne ako vo vete 3.13 vieme definovať dva binárne jazyky K a L akceptované m a n stavovými AFA tak, že minimálny AFA pre ich zreťazenie potrebuje $2^m + n + 1$ stavov. Týmto dostávame nasledujúce tvrdenie, ktoré rieši otvorený problém z [3, str. 131].

Veta 3.14 *Nech K a L sú jazyky nad abecedou Σ také, že $\text{asc}(K) = m$ a $\text{asc}(L) = n$, kde $m, n \geq 2$. Potom $\text{asc}(KL) \leq n + 2^m + 1$, a táto hranica je tesná, ak $|\Sigma| \geq 2$.*

3.4 Štvorec na booleovských automatoch

V tejto stati sa budeme venovať zreťazeniu a štvorcu na booleovských automatoch. Horný odhad stavovej zložitosti štvorca na booleovských automatoch priamo odvodíme zo stavovej zložitosti zreťazenia na booleovských automatoch, no zatiaľ sme pracovali len s alternujúcimi automatmi. Budeme však postupovať veľmi podobne a preto najskôr uvedieme lemu o zreťazení jazyka akceptovaného NNFA a jazyka akceptovaného BFA.

Ak $A = (Q, \Sigma, \delta, g_s, F)$ je BFA, kde $\delta(q, a)$ je konštanta alebo booleovská funkcia iba s $\vee$-operátorom pre každé $q \in Q$ a $a \in \Sigma$ a zároveň pre počiatočnú funkciu g_s platí, že je to disjunkcia stavov z Q, potom sa A správa ako NNFA. Hovoríme, že A je *NNFA v BFA reprezentácii*.

Lemma 3.15 *Nech $A_1 = (Q_1, \Sigma, \delta_1, g_1, F_1)$ je NNFA v BFA reprezentácii a nech $A_2 = (Q_2, \Sigma, \delta_2, g_2, F_2)$ je ľubovoľný BFA taký, že $Q_1 \cap Q_2 = \emptyset$. Uvažujme BFA $A_1 \cdot A_2 = (Q_1 \cup Q_2, \Sigma, \delta, g_1, F_2)$ taký, že pre všetky stavy $q \in Q_1 \cup Q_2$ a všetky písmená $a \in \Sigma$ je prechodová funkcia takáto:*

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & \text{ak } q \in Q_1 \text{ a } q \notin F_1; \\ \delta_1(q,a) \vee \delta_2(g_2,a), & \text{ak } q \in Q_1 \text{ a } q \in F_1; \\ \delta_2(q,a), & \text{ak } q \in Q_2. \end{cases}$$

Potom $L(A_1 \cdot A_2) = L(A_1) \cdot L(A_2)$.

Dôkaz.
Dôkaz tejto lemy prebieha takisto ako v Leme 3.11, kde sme zreťavovali jazyky akceptované NFA v AFA reprezentácii a AFA. Jediným rozdielom je to, že výpočet pre slovo $w \in L(A_1 \cdot A_2)$ nezačíname v jednom stave, ale počiatočnej booleovskej funkcii g_1, čo ale predstavuje podmnožinu stavov z Q_1.

□

Pre zreťazenie dvoch booleovských automatov automatov A_1 s m stavmi a A_2 s n stavmi využijeme predchádzajúcu lemu 3.15, no potrebujeme zreťaziť NNFA v BFA reprezentácii s booleovským automatom. Booleovský automat A_1 však vieme podľa vety 3.2 prerobiť na 2^m stavový NNFA. Dostávame teda nasledujúce tvrdenie o hornom odhade stavovej zložitosti zreťazenia na booleovských auomatoch.

Veta 3.16 *Nech A_1 a A_2 sú BFA s m a n stavmi. Potom existuje BFA A s najviac $2^m + n$ stavmi, ktorý akceptuje jazyk $L(A_1)L(A_2)$.*

Priamym dôsledkom tejto vety je horný odhad $2^n + n$ pre zložitosť štvorca na booleovských automatoch. Jeho tesnosť ukážeme v nasledujúcom tvrdení.

Veta 3.17 *Nech L je regulárny jazyk nad abecedou Σ taký, že $\mathrm{bsc}(L) = n$, kde $n \geq 2$. Potom $\mathrm{bsc}(L^2) \leq 2^n + n$, a táto hranica je tesná, ak $|\Sigma| \geq 2$.*

Dôkaz.
Tento dôkaz prebieha rovnako ako dôkaz vety 3.13, v ktorej sme ukázali tesnosť odhadu pre zložitosť štvorca na alternujúcich automatoch.

Nech L^R je jazyk akceptovaný DFA A z Obr. 2.1 s 2^n stavmi, kde polovica stavov ($k = 2^{n-1}$) je koncových. Z toho vyplýva, že L je akceptovaný n stavovým BFA. Vytvorme teraz štvorec pre náš jazyk L^R. Vidíme, že $(L^2)^R = (L^R)^2$. Spočítajme, koľko stavov by potreboval DFA pre takýto

jazyk: $\mathrm{sc}\big((L^R)^2\big) = (2^n - 2^{n-1}) \cdot 2^{2^n} + (2^{n-1}) \cdot 2^{2^n-1}$. Z toho vyplýva, že ak prejdeme k booleovským automatom, tak $\mathrm{bsc}(L^2) \geq \lceil \log\big((2^n - 2^{n-1}) \cdot 2^{2^n} + (2^{n-1}) \cdot 2^{2^n-1}\big) \rceil = 2^n + n$.

□

Ukázali sme tesné horné ohraničenia $2^n + n + 1$ a $2^n + n$ pre štvorec na alternujúcich a na booleovských automatoch. Všimnime si, že tieto hodnoty sa od seba líšia len o jednotku. Analógiou pre nás môže byť zjednotenie jazykov, ktoré sú zadané dvoma nedeterministickými automatmi s m a n stavmi. NNFA pre ich zjednotenie bude mať nanajvýš $m + n$ stavov, avšak NFA pre toto zjednotenie už bude prinajhoršom potrebovať $m+n+1$ stavov, lebo musíme mať len jeden počiatočný stav.

Záver

V tejto práci sme študovali operáciu štvorec na formálnych jazykoch reprezentovaných deterministickými, alternujúcimi a booleovskými konečnostavovými automatmi.

Najskôr sme sa venovali štvorcu na deterministických automatoch, ktoré mali n stavov, z ktorých bolo k koncových. Ukázali sme, že ak koncových stavov nie je priveľa, teda $1 \leq k \leq n-2$, tak minimálny deterministický automat pre štvorec má najviac $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ stavov. Pre každé také n a k sme našli binárny n stavový DFA s k koncovými stavmi o ktorom sme ukázali, že minimálny DFA pre jeho štvorec má práve $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ stavov. Toto je hlavný výsledok našej práce, ktorý sme v druhej časti použili na definovanie jazyka ťažkého pre štvorec na alternujúcich automatoch.

Následne sme sa venovali štvorcu na jazykoch, ktoré sú akceptované deterministickými automatmi s $n-1$ koncovými stavmi. Zistili sme, že v takom prípade minimálny DFA pre štvorec takýchto jazykov má vždy menej ako $(n-(n-1)) \cdot 2^n + (n-1) \cdot 2^{n-1} = (2n+2) \cdot 2^{n-2}$ stavov. Rozobrali sme dva prípady, a to keď v pôvodnom DFA pre L je počiatočný stav koncový alebo v pôvodnom DFA pre L je jediným nekoncovým stavom počiatočný stav.

V prvom prípade sme ukázali, že minimálny DFA pre štvorec takého jazyka môže mať najviac $(n+2) \cdot 2^{n-2}$ stavov a našli sme binárny DFA s $n-1$ koncovými stavmi, kde počiatočný stav je koncový o ktorom sme dokázali, že minimálny DFA pre jeho štvorec má presne toľko stavov.

V druhom prípade sme zistili, že vieme prekročiť hodnoty z predošlého prípadu ale nie o veľmi veľa. Počet stavov minimálneho deterministického automatu pre štvorec jazyka, kde len počiatočný stav je nekoncový sme ohraničili zhora vzťahom $(n+3) \cdot 2^{n-2}$. Našli sme ternárny jazyk vyhovujúci podmienkam, že minimálny DFA pre jeho štvorec potrebuje $(n+3) \cdot 2^{n-2}$ stavov. Len o jednotku menšiu hodnotu pre zložitosť štvorca vieme však dosiahnuť aj pre binárny automat. Naše výpočty ukázali, že túto hranicu nevieme v binárnom prípade presiahnuť pre $n \leq 5$. Či toto je pravda aj pre vyššie hodnoty n zostáva otvoreným problémom. Domnievame sa však, že túto hodnotu nemožno ani pre väčšie n v binárnom prípade presiahnuť.

V druhej časti práce sme sa venovali booleovským a alternujúcim automatom. Po úvode do problematiky sme sa zaoberali zreťazením alternujúcich automatov, kde sme ukázali, že ak jazyky K a L sú akceptované m a n stavovými alternujúcimi automatmi, tak potom ich zreťazenie KL možno akceptovať alternujúcim automatom, ktorý má najviac $2^m + n + 1$ stavov. Problém, či túto hodnotu možno dosiahnuť zostal v práci [3] otvorený.

Tento problém pozmenený pre štvorec bol našou motiváciou pre štúdium štvorca na deterministických automatoch. Na jeho vyriešenie sme využili nami nájdený binárny jazyk akceptovaný n stavovým deterministickým automatom s k koncovými stavmi, o ktorom sme predtým ukázali, že minimálny DFA pre jeho štvorec má $(n-k) \cdot 2^n + k \cdot 2^{n-1}$ stavov. Tento jazyk sme pozmenili tak, že sme mu dali 2^n stavov a polovica z nich bola koncových. Takto sme dostali binárny jazyk, ktorý je akceptovaný n stavovým alternujúcim automatom, pričom minimálny alternujúci automat pre jeho štvorec má $2^n + n + 1$ stavov. Pomocou takéhoto jazyka sme ukázali tesnosť horného odhadu $2^n + n + 1$ pre štvorec na alternujúcich automatoch. Zovšeobecním sme následne vyriešili otvorený problém tesnosti horného odhadu $2^m + n + 1$ pre zreťazenie formulovaný v [3]. Nakoniec sme ukázali, že zložitosť operácie štvorec na booleovských automatoch je $2^n + n$.

Zoznam použitej literatúry

[1] Brzozowski, J., Leiss, E.: On equations for regular languages, finite automata, and sequential networks. Theor. Comput. Sci. 10, 19–35 (1980)

[2] Čevorová, K., Jirásková, G., Krajňáková, I.: On the square of regular languages. In: Holzer, M., Kutrib, M. (eds.): CIAA 2014. LNCS, vol. 8487, pp. 136–147. Springer, Heidelberg (2014)

[3] Fellah, A., Jürgensen, H., Yu, S.: Constructions for alternating finite automata. Internat. J. Comput. Math. 35, 117–132 (1990)

[4] Jirásek, J., Jirásková, G., Szabari, A.: State complexity of concatenation and complementation. Int. J. Found. Com. Sci. 16, 511-529 (2005)

[5] Jirásková, G.: Descriptional complexity of operations on alternating and boolean automata. In: Hirsch, E. et al. (eds.) CSR 2012. LNCS, vol. 7353, pp. 196–204. Springer, Heidelberg (2012)

[6] Jirásková, G.: The ranges of state complexities for complement, star, and reversal of regular languages. Int. J. Found. Comput. Sci. 25, 101–124 (2014)

[7] Leiss, E.: Succinct representation of regular languages by Boolean automata. Theor. Comp. Sci. 13, 323–330 (1981)

[8] Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Dokl. 11, 1373–1375 (1970)

[9] Rampersad, N.: The state complexity of L^2 and L^k. Inf. Process. Lett. 98, 231–234 (2006)

[10] Sipser, M.: Introduction to the theory of computation. PWS Publishing Company (1997)

[11] Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some operations on regular languages. Theor. Comput. Sci. 125, 315–328 (1994)