ŠTRUKTURÁLNE VLASTNOSTI A STAVOVÁ ZLOŽITOSŤ UNÁRNYCH OPERÁCII NA KONEČNOSTAVOVÝCH AUTOMATOCH

autor: Bc. Šimon Javorský
vedúci: RNDr. Juraj Šebej, PhD.

Úvod a motivácia

- operačná stavová zložitosť
  - 1994
    - Yu, Zhuang, Salomaa, The state complexities of some basic operations on regular languages (1994)
  - 2020
    - Jirásek, Jirásková, Multiple concatenation and state complexity (2020)
  - 2020
    - Okhotin, Sazhneva, State complexity of gf(2)-inverse and gf(2)- star on binary languages (2020)

- modely
  - deterministický konečnostavový automat (DFA)
  - nedeterministický konečnostavový automat (NFA)
  - alternujúci automat (AFA)

- operácie
  - kleenov uzáver (Kleene star)
  - pozitívny uzáver (Plus)
  - štvorec (Square)

Stavová zložitosť

- stavová zložitosť jazyka L, sc(L)
- počet stavov najmenšieho DFA akceptujúceho L
  - „Pre každý jazyk existuje unikátny minimálny deterministický automat, ktorý ho akceptuje (až po izomorfizmus).“
    - Hopcroft, Motwani, Ullman, Introduction to Automata Theory, Languages, and Computation (2006)

- Príklad sc(L*) pre n=5
  - všetky jazyky nad unárnou abecedou so stavovou zložitosťou 5
  - všetky minimálne päťstavové unárne DFA
  - aplikácia Kleene star a minimalizácia
  - výsledkom sú automaty s počtom stavov 1 až 17 (s výnimkami)
  - rozsah zložitosti sc(L*) je $1 \leq \alpha \leq 17$

Stavová zložitosť

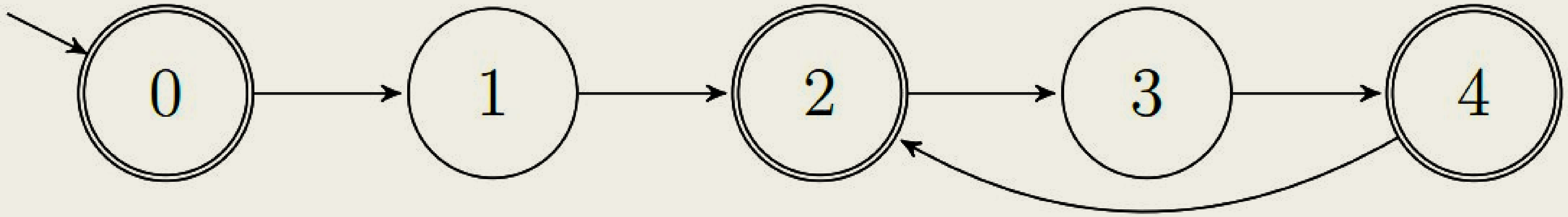
- problém zložitosti priemerného stavu
  - Nicaud, Average state complexity of operations on unary automata (1999)
- binárna a terárna abeceda a operácia štvorca
  - Krajňáková, Jirásková, Square on deterministic, alternating, and boolean finite automata (2017)
- existencia nedosiahnuteľných hodnôt z rozsahu stavových zložitostí
  - Geffert, Magic numbers in the state hierarchy of finite automata (2007)
- dve medzery nedosiahnuteľných hodnôt okolo n v rozsahu stavových zložitostí kleene star na unárnych jazykoch
  - Čevorová, Kleene star on unary regular languages (2013)

Stavová zložitosť

- obmedzenie na unárne automaty s polovičným počtom stavov akceptujúcich
  - „Jazyk je akceptovaný n-stavovým alternujúcim konečno stavovým automatom, práve vtedy keď je jeho reverz akceptovaný deterministickým konečno stavovým automatom s 2^n stavmi, pričom 2^{n-1} z nich sú akceptujúce."
    - Fellah, Jurgensen, Yu, Constructions for alternating finite automata (1990)
    - Hopcroft, Motwani, Ullman, Introduction to Automata Theory, Languages, and Computation (2006)
    - Jiraskova, Descriptional complexity of operations on alternating and boolean automata (2009)
    - Krajňáková, Jirásková, Square on deterministic, alternating, and boolean finite automata (2017)

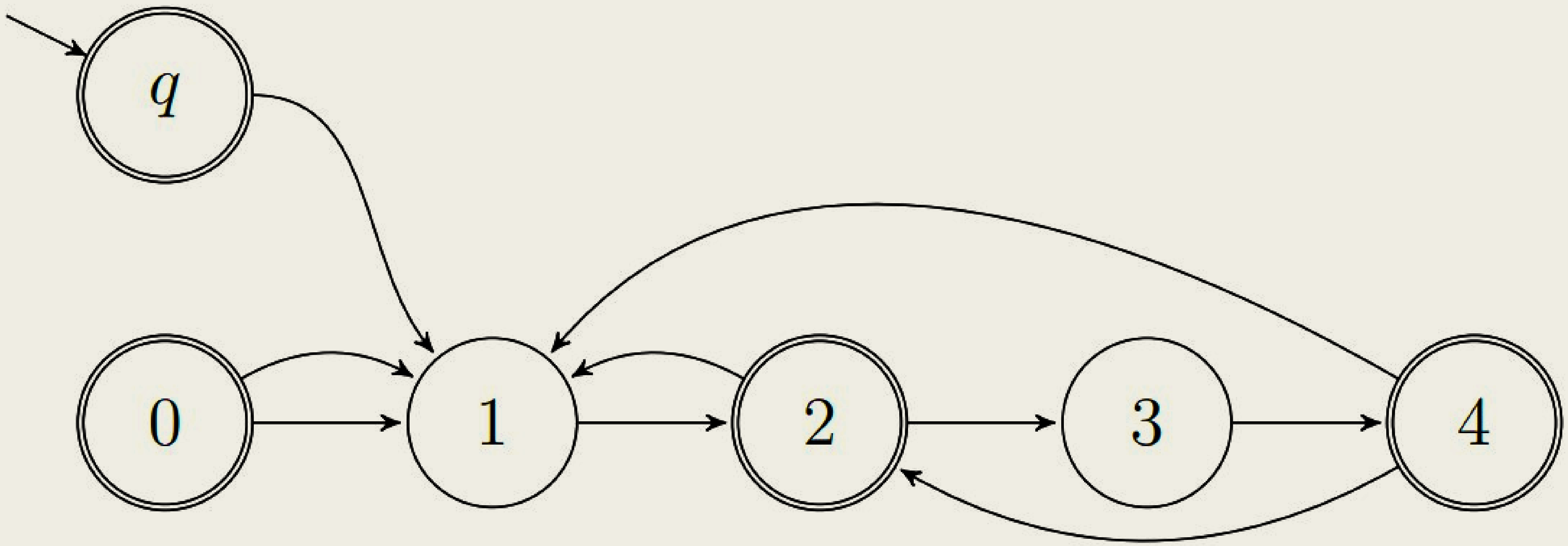
Príklad

- modifikovaná Nicaudová notácia
  - $\mathrm{M} = (Q, \Sigma, \delta, q_0, F)$
  - $\mathrm{M} = (n, l, F) = (5, 2, \{0, 2, 4\})$
  - $\mathrm{M} = (5{,}2{,}10101)$



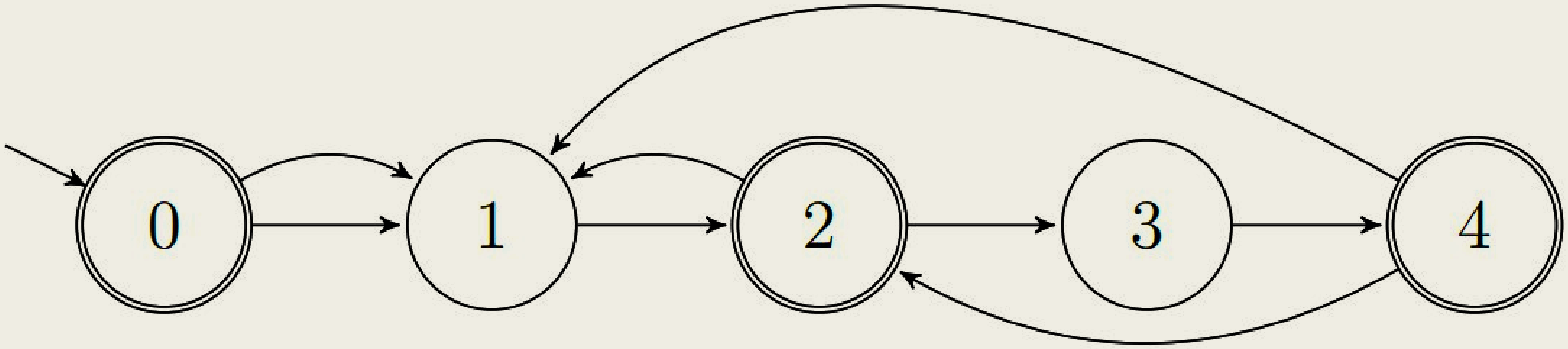
Príklad

- konštrukcia kleene star
  1. pridať nový akceptujúci stav q s prechodom na stav 1
  2. nastaviť stav q ako je jediný počiatočný
  3. pridať prechod z každého akceptujúceho stavu do stavu 1



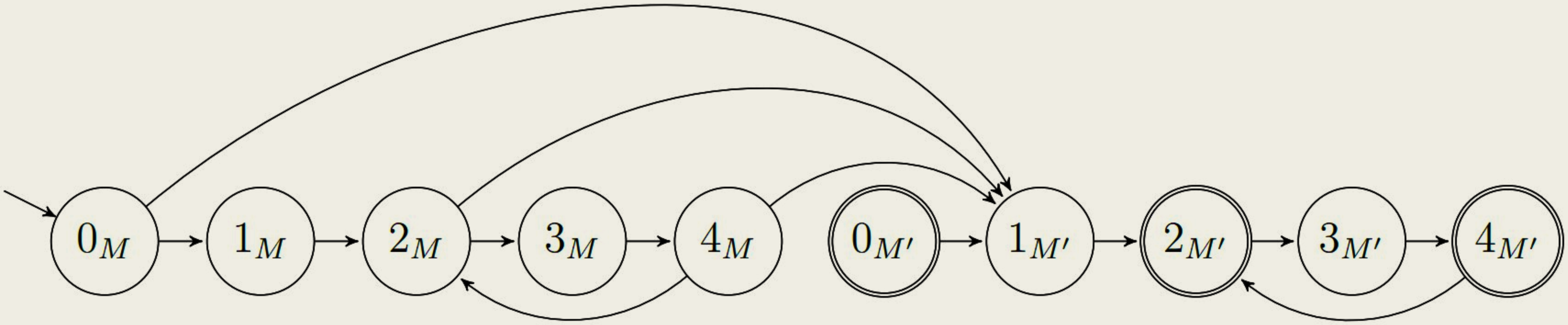
Príklad

- konštrukcia plus
    1. pridať prechod z každého akceptujúceho stavu do stavu 1



Príklad

- konštrukcia square
    1. vytvoriť kópiu M' automatu M
    2. pridať prechod z každého akceptujúceho stavu M do stavu $1_{M'}$
    3. nastaviť všetky stavy v M ako neakceptujúce



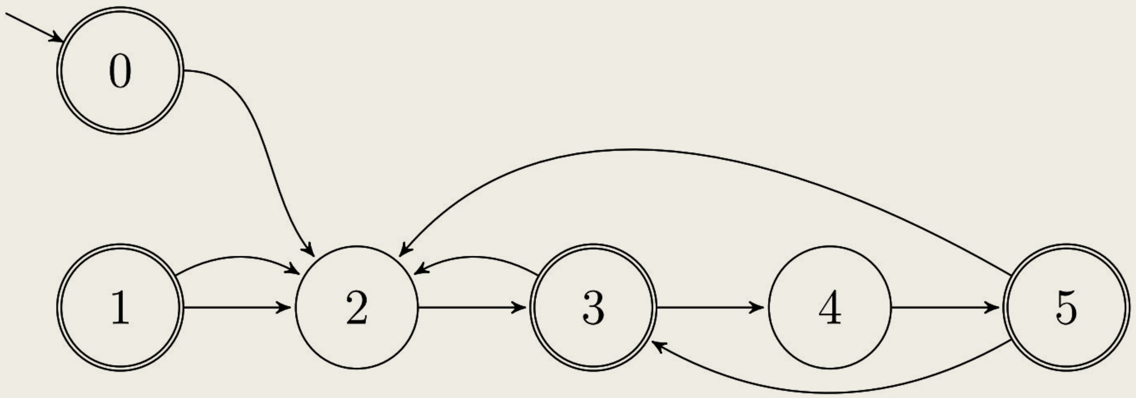
Príklad

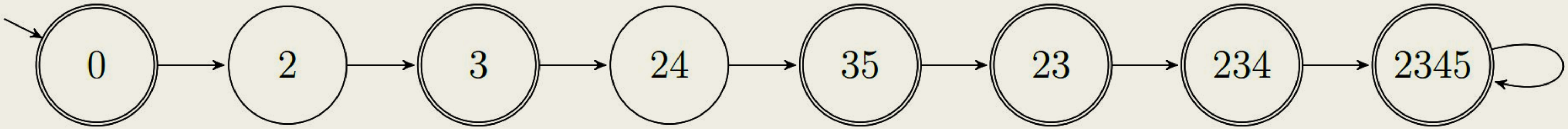
- konštrukcia podmnožín
  - z NFA $\mathrm{N} = (Q_N, \Sigma, \delta_N, q_0, F_N)$ na DFA $\mathrm{D} = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$
    - Q_D je množina podmnožín Q_N
    - F_D je množina podmnožín $S \in Q_N$ takých, že $\mathrm{S} \cap F_N \neq \emptyset$
    - pre každú množinu $S \in Q_N$ a pre každý vstupný symbol $\alpha \in \Sigma$ platí

$$\delta_D(S, \alpha) = \bigcup_{p \in S} \delta_N(p, \alpha)$$

Príklad

- konštrukcia podmnožín





Generovanie

- zvoľ veľkosť automatu *n*
- pre každé *l* od *0* po *n-1* opakuj
  - pre každé *stavy* z množiny všetkých kombinácií *0* a *1* dĺžky *n* opakuj
    - ak $stavy[l-1] \neq stavy[n-1]$
      - *stavySlucky* <- podzoznam stavy od *l* po *(n-1)*
      - ak *stavySlucky* je veľkosti prvočísla a obsahuje aj *0* aj *1* potom
        - spracuj automat *(n,l,stavy)*
      - inak
        - *M* <- automat *(n-l,0,stavySlucky)*
        - *M*‘ <- minimalizovaný *M* pomocou Hocroftovho algoritmu
        - ak *M* = *M*‘ potom
          - spracuj automat *(n,l,stavy)*

Generovanie

- "Unárny automat *(n,l,F)* je minimálny práve vtedy keď sú splnené obe podmienky:
  1. jeho slučka je minimálna,
  2. stavy *(n-1)* a *(l-1)* nemajú rovnakú finalitu."
     - Nicaud, Average state complexity of operations on unary automata (1999)

- "Majme p-stavový automat *M*, kde *p* je prvočíslo, navyše *M* spĺňa $\delta(i,\alpha) = i + 1 \; mod \; p$ pre všetky $i \in Q$. Ak existuje stav $j \in F$ a stav $k \notin F$, potom je automat M minimálny."
  - Hopcroft, Motwani, Ullman, Introduction to Automata Theory, Languages, and Computation (2006)

Spracovanie

- $M_1 \leftarrow$ nedeterministický automat získaný aplikovaním vybranej operácie
- $M_2 \leftarrow$ deterministický automat získaný aplikovaním konštrukcie podmnožín na M_1
- $M_3 \leftarrow$ minimálny automat získaný aplikovaním Hopcroftovho algoritmu na M_2
- $m \leftarrow \max\{\text{veľkosti vo } f\}$
- ak $|M_3| \in f$ potom
  - ak $\mathrm{m} = |M_3|$ potom
    - pridať M_3 k príkladom $\mathrm{f}[m]$
  - zvýšiť počítadlo $\mathrm{f}[|M_3|]$
- inak
  - ak $\mathrm{m} < |M_3|$ potom
    - odstrániť všetky príklady z $\mathrm{f}[m]$ okrem prvého
  - pridať $|M_3|$ do f s počítadlom 1 a príkladom M_3

Spracovanie

- maximálne dosiahnuté stavové zložitosti
- vzorce v stavových zložitostiach
- absencia konkrétnych stavov
- počty vygenerovaných automatov
- štruktúry vygenerovaných automatov
- iné zaujímavé úkazy

Záver

- Otázky?