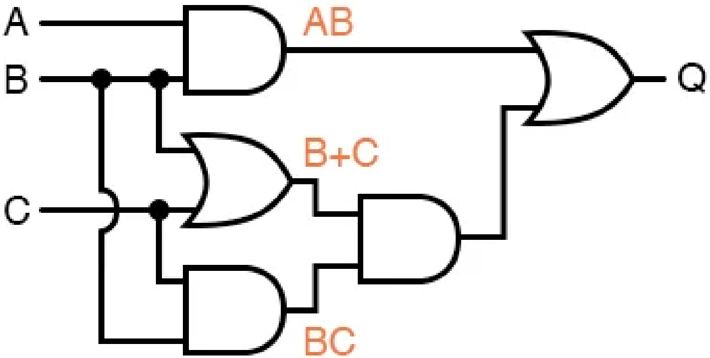
Partially nondeterministic automata - Nondeterministic choice of initial states

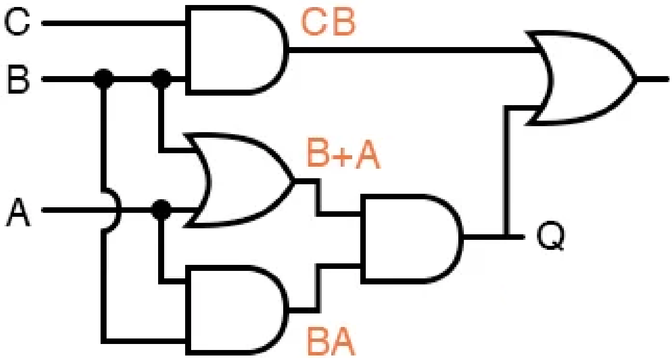
Šimon Huraj
Tutor: RNDr. Juraj Šebej, PhD.

Bachelor's thesis defense
June 20, 2023

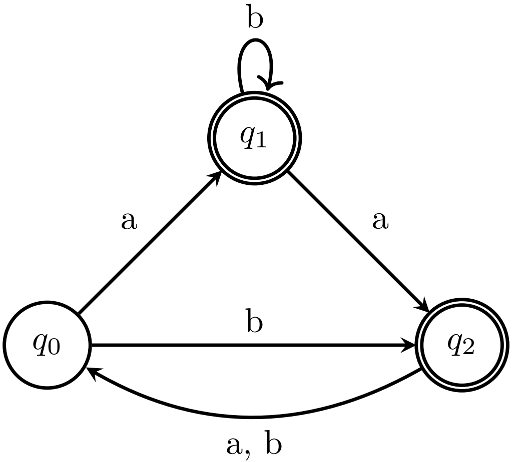
Motivation



Motivation



Motivation



Deterministic finite-state automaton (DFA)

Definition[1]

Deterministic finite-state automaton (DFA) is a quintuple $A = (Q, \Sigma, q_0, F, \delta)$ where

- Q is a finite set of states
- Σ is a finite set of input symbols
- q_0 is a initial state, $q_0 \in Q$
- σ is a transition function, $\sigma : Q \times \Sigma \to Q$
- F is a set of final states, $F \subseteq Q$

[1] (J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to automata theory, languages, and computation.* Vol. 2. Addison-Wesley, 2003. ISBN: 978-0201441246)

k-entry Deterministic finite-state automaton ($_kDFA$)

Definition[2]

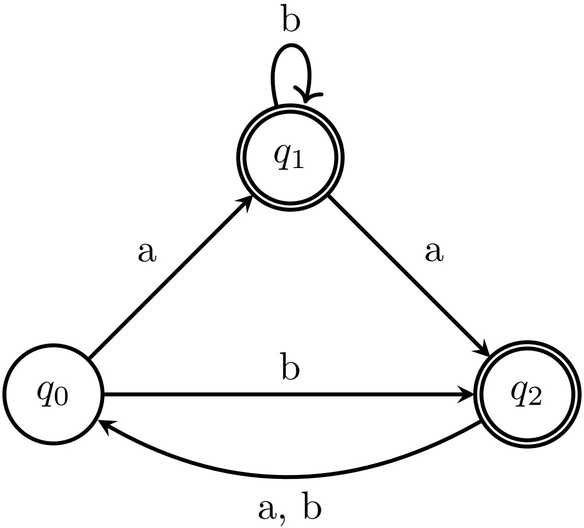
k-entry deterministic finite-state automaton ($_kDFA$) is a quintuple $M = (Q, \Sigma, I, F, \delta)$ where

- Q is a finite set of states
- Σ is a finite set of input symbols
- **I is a set of initial states, $I \subseteq Q$**
- σ is a transition function, $\sigma : Q \times \Sigma \to Q$
- F is a set of final states, $F \subseteq Q$

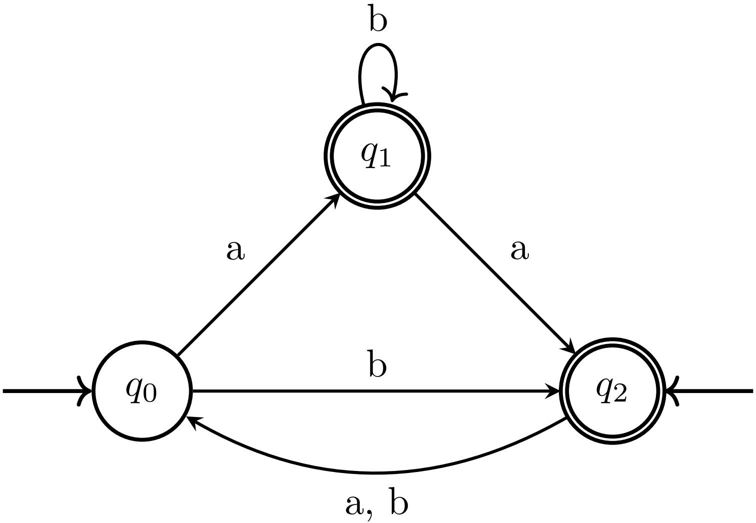
Note: The only point of non-determinism is in choice of initial states

[2] (M. Holzer, K. Salomaa, and S. Yu. "On the State Complexity of k-Entry Deterministic Finite Automata". In: *Journal of Automata, Languages and Combinatorics* 6.4 [2001], pp. 453–466)

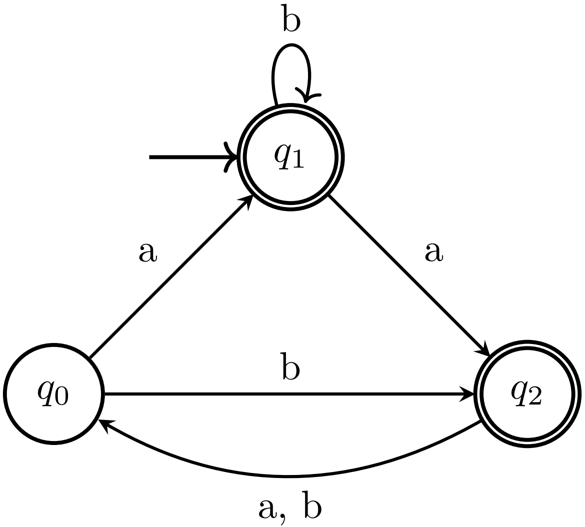
${}_kDFA$ example



${}_kDFA$ example



${}_{k}DFA$ example

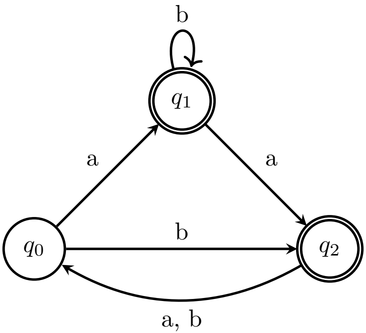


Family

Conversion of three state family to number

122100 FTT $\rightarrow (122100)_3 \quad (011)_2$

transitions final states

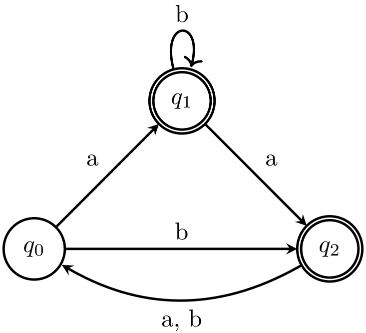


Family

Conversion of three state family to number

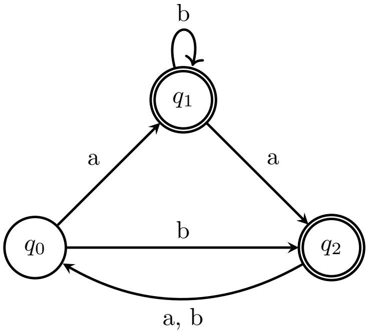
122100 FTT $\rightarrow (122100)_3 \quad (011)_2 \rightarrow (111010100)_2 \quad (011)_2 \rightarrow (111010100011)_2 = 3747_{10}$

transitions final states



Generation

Family number

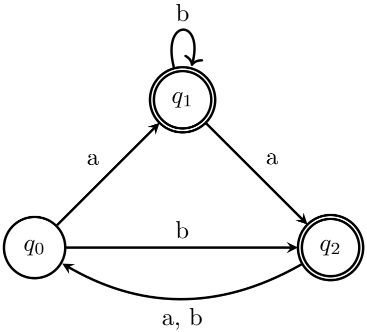


Generation

Family number

initial states	state complexity	language group
000		
001		
010		
011		
100		
101		
110		
111		

2^n
automata

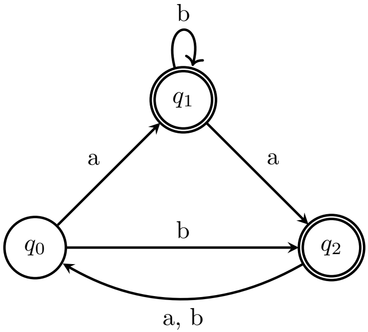


Generation

Family number

initial states	state complexity	language group
000		
001		
010		
011		
100		
101		
110		
111		

$\underset{\text{automata}}{2^n} \quad \cdot(\underset{\text{unreach.}}{n}$

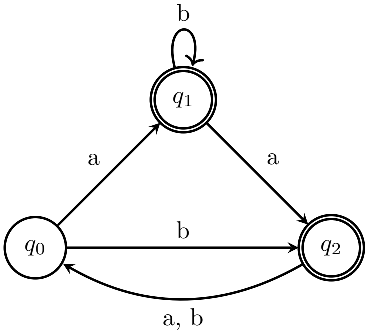


Generation

Family number

initial states	state complexity	language group
000		
001		
010		
011		
100		
101		
110		
111		

$\underset{\text{automata}}{2^n} \cdot (\underset{\text{unreach.}}{n} + \underset{\text{det.}}{2^n}$

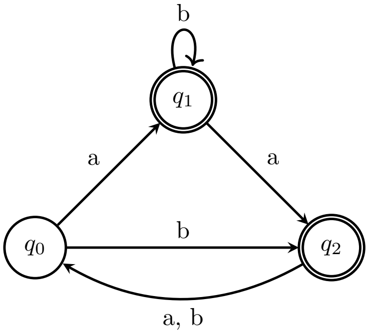


Generation

Family number

initial states	state complexity	language group
000		
001		
010		
011		
100		
101		
110		
111		

$$\underbrace{2^n}_{\text{automata}} \cdot (\underbrace{n}_{\text{unreach.}} \underbrace{+2^n}_{\text{det.}} \underbrace{+2^n \cdot n}_{\text{minim.}})$$

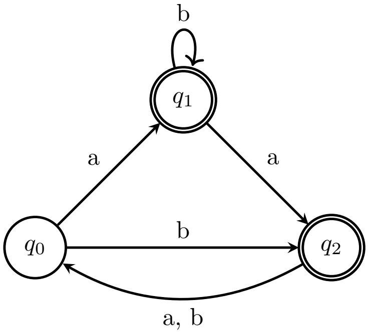


Generation

Family number

initial states	state complexity	language group
000	1	
001	3	
010	3	
011	1	
100	3	
101	1	
110	1	
111	1	

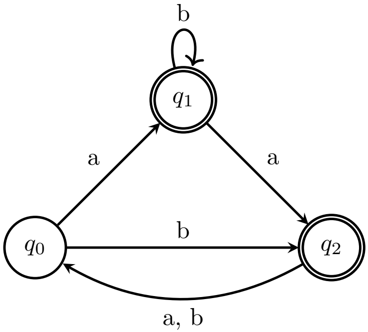
$$\underset{\text{automata}}{2^n} \cdot (\underset{\text{unreach.}}{n} + \underset{\text{det.}}{2^n} + \underset{\text{minim.}}{2^n \cdot n})$$



Generation

Family number

initial states	state complexity	language group
000	1	
001	3	
010	3	
011	1	
100	3	
101	1	
110	1	
111	1	

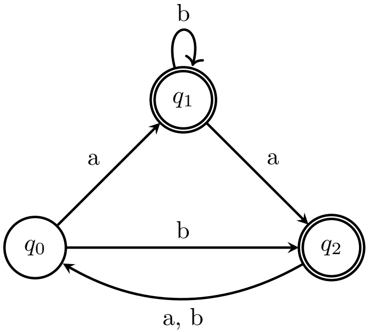


$$\underbrace{2^n}_{\text{automata}} \cdot (\underbrace{n}_{\text{unreach.}} + \underbrace{2^n}_{\text{det.}} + \underbrace{2^n \cdot n}_{\text{minim.}}) \cdot \underbrace{2^n 2^{n-2}}_{\text{choice of pair}}$$

Generation

Family number

initial states	state complexity	language group
000	1	
001	3	
010	3	
011	1	
100	3	
101	1	
110	1	
111	1	



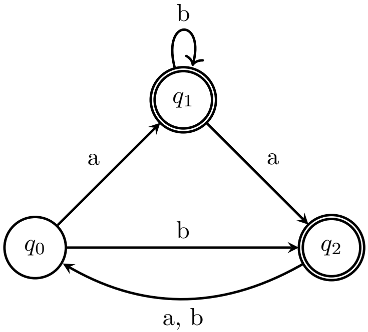
$$\underbrace{2^n}_{\text{automata}} \cdot (\underbrace{n}_{\text{unreach.}} + \underbrace{2^n}_{\text{det.}} + \underbrace{2^n \cdot n}_{\text{minim.}}) \cdot \underbrace{2^n 2^{n-2}}_{\text{choice of pair}} \cdot \underbrace{2^{2n} 2n}_{\text{equivalence}}$$

Generation

Family number

initial states	state complexity	language group
000	1	A
001	3	B
010	3	C
011	1	D
100	3	E
101	1	D
110	1	D
111	1	D

Number of distinct languages: 5

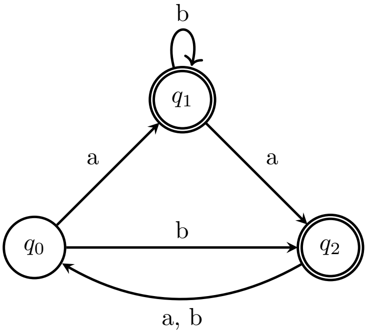


$$\underbrace{2^n}_{\text{automata}} \cdot (\underbrace{n}_{\text{unreach.}} + \underbrace{2^n}_{\text{det.}} + \underbrace{2^n \cdot n)}_{\text{minim.}} \cdot \underbrace{2^n 2^{n-2}}_{\text{choice of pair}} \cdot \underbrace{2^{2n} 2n}_{\text{equivalence}}$$

up to $n \leq 4$

Generation

Family number

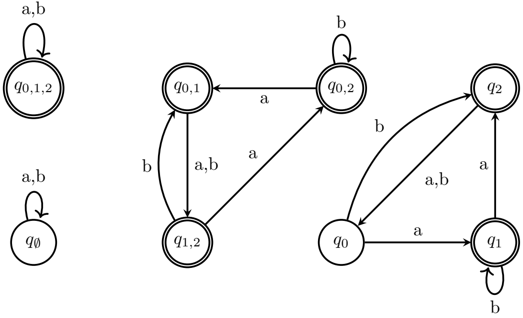


Generation

Family number

states in subset	language group	state complexity
000		
001		
010		
011		
100		
101		
110		
111		

2^n
determinization

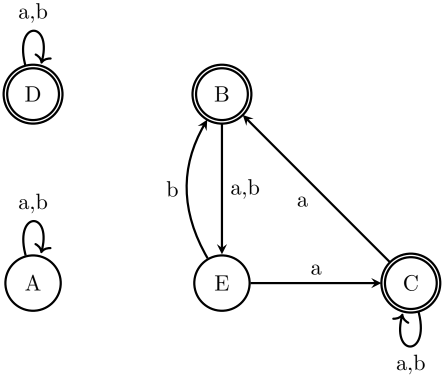


Generation

Family number

states in subset	language group	state complexity
000	A	
001	B	
010	C	
011	D	
100	E	
101	D	
110	D	
111	D	

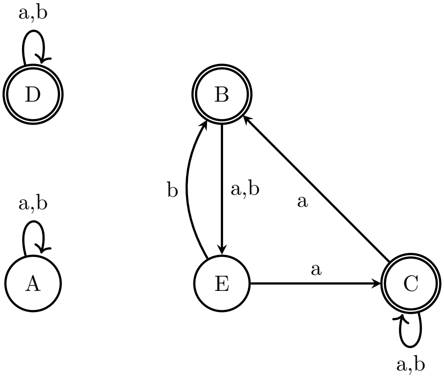
$$\underbrace{2^n}_{\text{determinization}} + \underbrace{2^n n}_{\text{minimization}}$$



Generation

Family number

states in subset	language group	state complexity
000	A	
001	B	
010	C	
011	D	
100	E	
101	D	
110	D	
111	D	

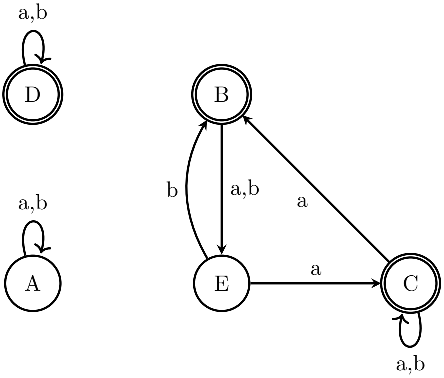


$$\underset{\text{determinization}}{2^n} + \underset{\text{minimization}}{2^n n} + \underset{\text{initial state}}{2^n}$$

Generation

Family number

states in subset	language group	state complexity
000	A	
001	B	
010	C	
011	D	
100	E	
101	D	
110	D	
111	D	

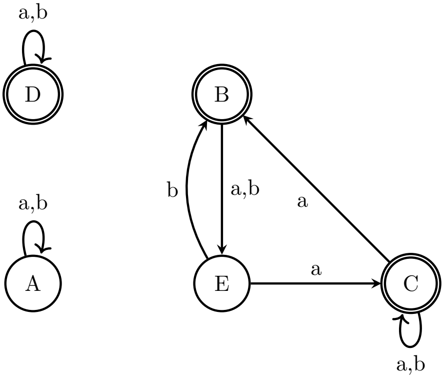


$$\underset{\text{determinization}}{2^n} + \underset{\text{minimization}}{2^n n} + \underset{\text{initial state}}{2^n} \cdot \underset{\text{reachability}}{2^n}$$

Generation

Family number

states in subset	language group	state complexity
000	A	1
001	B	3
010	C	3
011	D	1
100	E	3
101	D	1
110	D	1
111	D	1



$$\underset{\text{determinization}}{2^n} + \underset{\text{minimization}}{2^n n} + \underset{\text{initial state}}{2^n} \cdot \underset{\text{reachability}}{2^n}$$

up to $n \leq 5$,

up to $n \leq 6$ (in 4 - 6 weeks)

Generation Comparison

$$\boxed{2^n \cdot (n + 2^n + 2^n \cdot n) + 2^n \cdot 2^{n-2} \cdot 2^{2n} \cdot 2n} \text{ vs. } \boxed{2^n + 2^n \cdot n + 2^n \cdot 2^n}$$

- $\mathcal{O}(2^{4n-1}n)$ vs. $\mathcal{O}(2^n n)$ for number of distinct languages
- $\mathcal{O}(2^{2n}n)$ vs. $\mathcal{O}(2^{2n})$ for state complexity of languages

Computational results of average state complexity

n	average s.c. (computations)
2	1.29
3	2.09
4	3.63
5	6.20
6	10.09
7	15.70

Note: Alphabet of size 2.

Average state complexity

Lemma 1[3]

The minimal deterministic finite automaton accepting $L(M_{l,n})$ has $\sum_{i=1}^{l} \binom{n}{i}$ states.

Note: The size of alphabet does not play role in this formula.

Theorem 1

Average state complexity of a language represented by an n-state ${}_kDFA$ is at most

$$\frac{\sum_{i=1}^{n} \left(\binom{n}{i} \cdot \sum_{j=1}^{i} \binom{n}{j} \right)}{\sum_{i=1}^{n} \binom{n}{i}}$$

[3] (M. Holzer, K. Salomaa, and S. Yu. "On the State Complexity of k-Entry Deterministic Finite Automata". In: *Journal of Automata, Languages and Combinatorics* 6.4 [2001], pp. 453–466)

Average state complexity

n	average s.c. (computations)	average s.c. (formula)	2^n (for comparison)
2	1.29	2.34	4
3	2.09	4.86	8
4	3.63	9.80	16
5	6.20	19.55	32
6	10.09	38.83	64
7	15.70	77.01	128

Note: Alphabet of size 2 (computations).
Does not depend on alphabet size (formula).

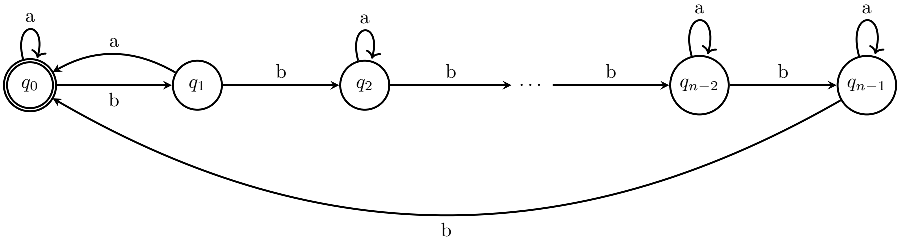
Theorem 2

Average state complexity of a language represented by an n-state $_k DFA$ is at most $\frac{5}{8}2^n$.

Family with maximum languages

Lemma 3

For every $n \geq 2$ exists n-state family such that, all the languages of the family are pairwise distinguishable.

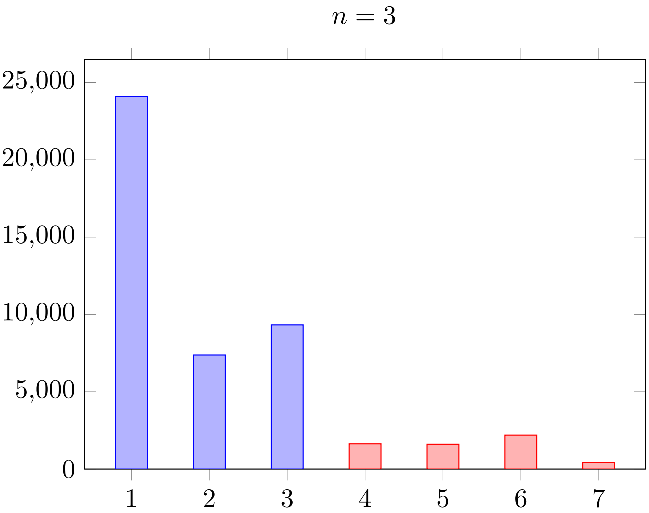


Summary and future work

- ${}_kDFA$ is a DFA except every state can be initial, similar as electrical (logical) circuit we can start it with different configurations
- better overall construction ($\mathcal{O}(2^{4n-1}n)$ vs. $\mathcal{O}(2^{2n})$)
- computational results - average s.c., distinct languages, number of distinct s.c.
- formal result - average s.c. exact formula, and other results
- existence of family with maximum number of languages for every n

Summary and future work

n	average s.c. (computations)	average s.c. (formula)
2	1.29	2.34
3	2.09	4.86
4	3.63	9.80
5	6.20	19.55
6	10.09	38.83
7	15.70	77.01



Thank you for your attention

Questions

Posledná veta kapitoly 1.1: Vysvetlite prečo ste sa rozhodli neodstraňovať nedosiahnuteľné stavy. Má to nejaký vplyv na zistené priemerné zložitosti?

as described in [6], has been adopted with minor modifications. Notably, during the employment of the complete subset construction methodology, no *unreachable* state is being removed.

Questions

Očakávate nejaké zmeny v priemernej zložitosti pre jazyky nad inou ako binárnou vstupnou abecedou?

states\alphabet size	**1**	**2**	**3**	**4**	**5**
2	1,15	1,39	1,42	1,45	1,47
3	1,34	2,25	2,69	3,06	
4	1,50	3,81			

Questions

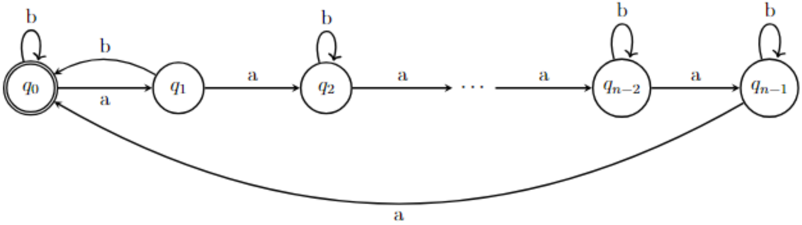
Strana 33, Veta 4.3: Mal by byť nejaký zásadný rozdiel pre priemernú stavovú zložitosť v prípade, že je n nepárne, alebo rozdiel oproti párnemu n bude iba typu „±konštanta“?

Theorem 4.3 *Let $n \geq 5$ be odd number, then average state complexity of a language represented by an n-state ${}_k$DFA is at most $5/8 \times 2^n$.*

n	actual average s.c.	average s.c. computed by formula	approximation(2^n)
2	1.39	2.34	4
3	2.25	4.86	8
4	3.81	9.80	16
5	6.37	19.55	32
6	10.24	38.83	64
7	15.81	77.01	128

Questions

Poznámka: Vetu 4.4 je možné veľmi ľahko zovšeobecniť z binárnych regulárnych jazykov na regulárne jazyky s ľubovoľnou inou vstupnou abecedou, vrátane unárnej.



Questions

Ako to vyzerá s priemernou stavovou zložitosťou pre deterministické rodiny automatov – t.j., ak nemáme pre danú rodinu automatov na výber celú podmnožinu počiatočných stavov ale len jediný počiatočný stav?

Lemma 4.1 *[5] The minimal deterministic finite automaton accepting $L(M_{l,n})$ has $\sum_{i=1}^{l} \binom{n}{i}$ states.*