P. J. Safarik University

Faculty of Science

PARTIALLY NONDETERMINISTIC AUTOMATA - NONDETERMINISTIC CHOICE OF INITIAL STATES

BACHELOR'S THESIS

Field of Study:	**Informatics**
Institute:	**Institute of Computer Science**
Tutor:	**RNDr. Juraj Šebej, PhD.**

Košice 2023

Šimon Huraj

Introduction

Even though automata theory is well-researched, there are still many unanswered questions. As many previous papers have stated state complexity is a natural complexity measure for any kind of automaton, therefore we decided to provide additional insights into this topic.

In this study we consider specific class of automata, that have an inner structure deterministic but the choice of initial states is nondeterministic, meaning that any state can serve as an initial. This type of automaton was already studied in [1] where they call it k-entry DFA. In their work, they proved, among other things, the upper bound for state complexity. This thesis focuses more on the average state complexity and also the resulting state complexity based on the choice of the initial states.

We also study groups of automata that are similar in structure but differ in their set of initial states.

Chapter 1

Definitions and Notation

It is assumed that the reader has a basic understanding of automata theory as outlined in Hopcroft and Ullman's work [2].

Definition 1.1 Let $k \geq 1$. A k—entry deterministic finite automaton (${}_k$DFA) is a quintuple $M = (Q, \Sigma, \sigma, I, F)$, where Q is a finite set of states, Σ a finite set of input symbols, $\sigma : Q \times \Sigma \to Q$ is the transition function, $I \subseteq Q$ is the set of initial states with $|I| = k$, and $F \subseteq Q$ the set of final states [1].

Definition 1.2 Let $k \geq 1$. Family of finite automata will represent a group of ${}_k$DFAs where all the automata in the particular group have equivalent:

- set of states Q,
- set of input symbols Σ,
- transition function σ,
- set of final states F,

they only differ in the set of initial states I

Chapter 2

Java Application

Since working with many automata, sometimes even millions, by hand would be impossible, we decided to create a simple application that will handle this issue for us and will help us analyze whole groups of automata at once. This application is capable of generating automata, either one specific or range of more, as well as a few basic operations with them.

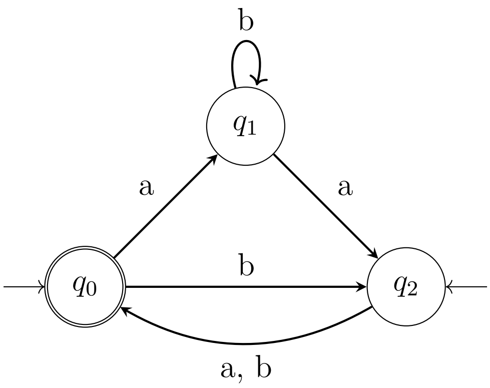
2.1 Automaton Representation

Although people prefer the graphical interpretation of automaton for a better understanding of its structure, it is more suitable for computers to work with numbers. That is the first reason why we opted for this representation. The second is that it is easier for us to generate a successor for chosen automaton simply by adding 1 to the already generated number.

To transform automaton to number representation, firstly we need to rename states from 0 to $n-1$ if they are not already. Then the process is relatively straightforward. Let $|Q| = n$ and $|\Sigma| = k$. Firstly, we will construct a number in the base-n numeral system having $k.n$ cyphers. Let $i \in \{0, ..., n\}$ and $j \in \{0, ..., k\}$, then $(i{+}j)$-th cipher represents destination state for q_i reading j-th letter of alphabet. Converted to base-10 we will call it *Transitions* number. Secondly, create *FinalStates* number, which is a binary number with n cyphers, the value of i-th one depends on whether q_i is final or not. Next, construct *InitialStates* number, similar to *FinalStates* number, it is a binary number with n cyphers. The value of i-th cypher depends on whether q_i is initial or not. Finally, convert *FinalStates* number and *InitialStates* number to base-10 and add all the numbers together by the formula:

$$(Transitions * 2^n + FinalStates) * 2^n + InitialStates$$

Example 2.1 (Conversion) Let's have a automaton M=(Q, Σ, σ, q_0, F), Q=$\{q_0, q_1, q_2\}$, Σ=$\{$a,b$\}$, F=$\{q_0\}$ and transitions are as shown below



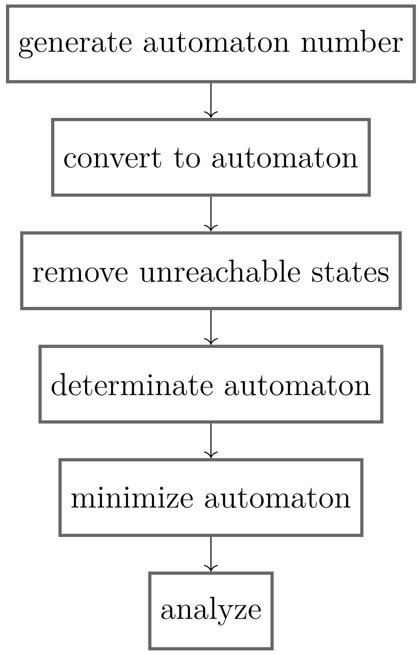
Considering $|Q| = n = 3$ and $|\Sigma| = k = 2$, then $Transitions = (122100)_3, FinalStates = (100)_2$ and $InitialStates = (101)_2$. Converting numbers and putting it all together we get

$$(468 * 2^3 + 4) * 2^3 + 5 = 29989$$

and that is a corresponding number to automaton M.

2.2 Generating Automata - Data set

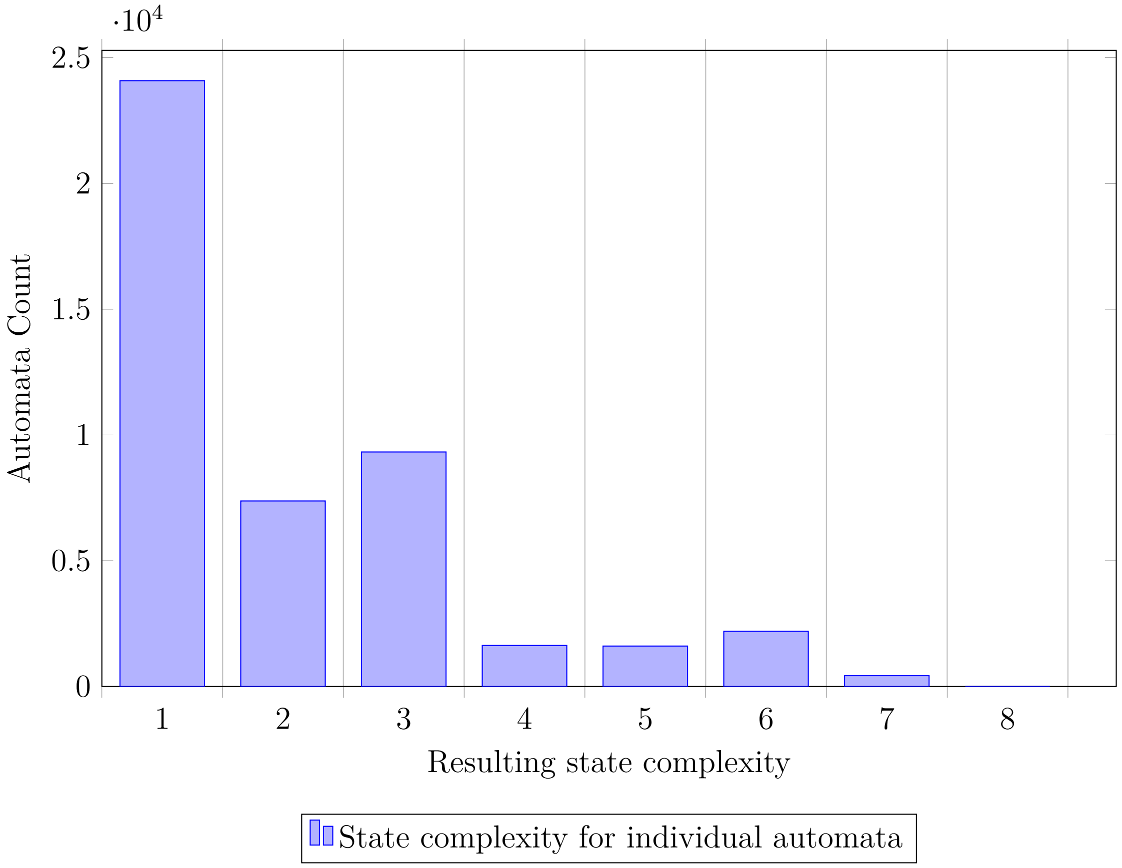
We use our simple Java application to generate the entire range of numbers representing some unique automaton which initially had a set of states the size of our chosen constant. Then the program converts the generated numbers to automata, performs some basic algorithms on them and does an analysis of the results either for one separate automaton, one complete family or all the generated automata. The complete process can be seen in the diagram below:



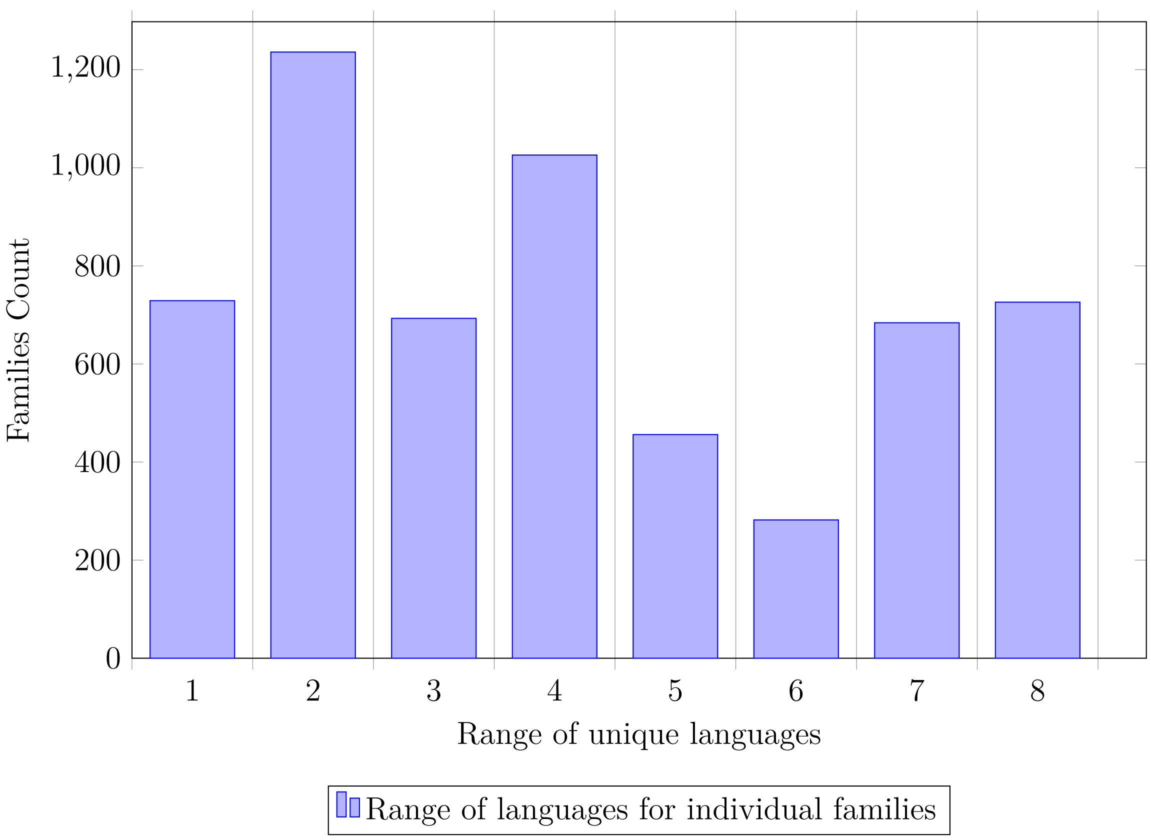
2.3 Results of an analysis

Our analysis mainly focuses on the state complexity of individual automata after determining and minimising, the range of languages that one family can represent and the range of state complexity one family can have. Although the process of generating and analyzing is fairly simple, it is computationally heavy, therefore we were able to acquire results only for automata that had initially set of states the size less than or equal to 5.

Results for different set sizes are more or less equal, thus we will only provide plots for only automata that initially had a set of states of size 3.



$\cdot 10^4$
2.5
2
1.5
1
0.5
0
Automata Count
1
2
3
4
5
6
7
8
Resulting state complexity
State complexity for individual automata



1,200
1,000
800
600
400
200
0
Families Count
1
2
3
4
5
6
7
8
Range of unique languages
Range of languages for individual families

Chapter 3

Average State Complexity

In this paper, we went a step further than Holzer, Salomaa and Yu in [1], where they only provided an upper bound for state complexity of ${}_kDFA$, and tried to come up with a way to determine average state complexity for this type of automata. We have used their upper bound to derive an equation for average state complexity.

3.1 Equation

$$\begin{aligned}\frac{\sum_{i=1}^{n}\left(\binom{n}{i}\cdot\sum_{j=1}^{i}\binom{n}{j}\right)}{\sum_{i=1}^{n}\binom{n}{i}} &= \frac{\sum_{j=1}^{1}\binom{n}{1}\binom{n}{j}+\sum_{j=1}^{2}\binom{n}{2}\binom{n}{j}+\cdots+\sum_{j=1}^{n}\binom{n}{n}\binom{n}{j}}{\sum_{i=1}^{n}\binom{n}{i}} = \\ &= \frac{\binom{n}{1}\cdot\sum_{j=1}^{1}\binom{n}{j}+\binom{n}{2}\cdot\sum_{j=1}^{2}\binom{n}{j}+\cdots+\binom{n}{n}\cdot\sum_{j=1}^{n}\binom{n}{j}}{\sum_{i=1}^{n}\binom{n}{i}}\end{aligned} \tag{3.1}$$

now we have to distinguish between two cases when n is odd or even. Let us first discuss a first case when n is odd:

$$\begin{aligned}&= \frac{\binom{n}{1}\cdot\left(\sum_{j=1}^{1}\binom{n}{j}+\sum_{j=1}^{n-1}\binom{n}{j}\right)+\cdots+\binom{n}{\frac{n-1}{2}}\cdot\left(\sum_{j=1}^{\frac{n-1}{2}}\binom{n}{j}+\sum_{j=1}^{\frac{n+1}{2}}\binom{n}{j}\right)}{\sum_{i=1}^{n}\binom{n}{i}}+1 = \\ &= \frac{\sum_{j=1}^{\frac{n-1}{2}}\left(\binom{n}{j}\left(2^n-2+\binom{n}{j}\right)\right)}{2^n-1}+1 = \frac{(2^n-2)\cdot\sum_{j=1}^{\frac{n-1}{2}}\binom{n}{j}+\sum_{j=1}^{\frac{n-1}{2}}\binom{n}{j}^2}{2^n-1}+1 = \\ &= \frac{(2^n-2)\cdot(2^{n-1}-1)+\frac{\binom{2n}{n}}{2}-1}{2^n-1}+1 = \frac{(2^n-2)^2+\binom{2n}{n}-2}{2\cdot(2^n-1)}+1,\end{aligned} \tag{3.2}$$

we obtain this result using operations with binomial coefficients and by equations:

$$\sum_{i=1}^{\frac{n-1}{2}}\binom{n}{i} = \frac{2^n}{2}-1 = 2^{n-1}-1,$$

$$\sum_{i=1}^{\frac{n-1}{2}}\binom{n}{i}^2 = \frac{\binom{2n}{n}}{2}-1$$

which are valid when n is odd, which is our case.

The second case when n is even:

$$= \frac{\binom{n}{1}\left(\sum_{j=1}^{1}\binom{n}{j} + \sum_{j=1}^{n-1}\binom{n}{j}\right) + \cdots + \binom{n}{\frac{n}{2}-1}\left(\sum_{j=1}^{\frac{n-1}{2}}\binom{n}{j} + \sum_{j=1}^{\frac{n+1}{2}}\binom{n}{j}\right) + \binom{n}{\frac{n}{2}}\sum_{j=1}^{\frac{n}{2}}\binom{n}{j}}{\sum_{i=1}^{n}\binom{n}{i}} + 1 =$$

$$= \frac{\sum_{j=1}^{\frac{n}{2}-1}\left(\binom{n}{j}\left(2^n - 2 + \binom{n}{j}\right)\right) + \binom{n}{2}\cdot\sum_{j=1}^{\frac{n}{2}}\binom{n}{j}}{2^n - 1} + 1 =$$

$$= \frac{(2^n - 2)\cdot\sum_{j=1}^{\frac{n}{2}-1}\binom{n}{j} + \sum_{j=1}^{\frac{n}{2}-1}\binom{n}{j}^2 + \binom{n}{2}\cdot\sum_{j=1}^{\frac{n}{2}}\binom{n}{j}}{2^n - 1} + 1 =$$

$$= \frac{\left(2^n - 2 + \binom{n}{\frac{n}{2}}\right)\cdot\sum_{j=1}^{\frac{n}{2}-1}\binom{n}{j} + \sum_{j=1}^{\frac{n}{2}-1}\binom{n}{j}^2 + \binom{n}{\frac{n}{2}}^2}{2^n - 1} + 1 =$$

$$= \frac{\left(2^n - 2 + \binom{n}{\frac{n}{2}}\right)\cdot\left(2^{n-1} - \binom{n-1}{\frac{n}{2}} - 1\right) + \frac{\binom{2n}{n}}{2} - 2\binom{n-1}{\frac{n}{2}}^2 - 1 + \binom{n}{\frac{n}{2}}^2}{2^n - 1} + 1, \tag{3.3}$$

again we used operations with binomial coefficients and equations:

$$\sum_{i=1}^{\frac{n}{2}-1}\binom{n}{i} = \frac{2^n}{2} - 1 - \binom{n-1}{\frac{n}{2}} = 2^{n-1} - \binom{n-1}{\frac{n}{2}} - 1$$

$$\sum_{i=1}^{\frac{n}{2}-1}\binom{n}{i}^2 = \frac{\binom{2n}{n} - \left(2\binom{n-1}{\frac{n}{2}}\right)^2}{2} - 1 = \frac{\binom{2n}{n}}{2} - 2\binom{n-1}{\frac{n}{2}}^2 - 1,$$

which are valid when n is even and that is the case.

Using *Equation 3.2* and *Equation 3.3* we gain an equation for calculating average worst-case state complexity which does not depend on parity of n.

After further investigation and approximation, we came to the conclusion that this approach for calculating an average state complexity is unsatisfactory because we are getting the worst possible scenario which is $2^n - 1$.

Conclusion

In this paper, we have laid down definitions needed for further understanding of a problem we examine. We have introduced our application for working with automata in a big scope and explained automaton representation with an algorithm for converting between number and quintuple $(Q, \Sigma, \sigma, I, F)$

We have also presented some results that we obtained through our application and which will be analyzed more thoroughly in future work.

We have also done some research on average state complexity that was mentioned in the Introduction, but this paper does not contain any more details, because they yet have to be formalized.

Bibliography

[1] Holzer, M., Salomaa, K. and Yu, S., 2001. On the State Complexity of k-Entry Deterministic Finite Automata. In: *Journal of Automata, Languages and Combinatorics*. Vol. 6, no. 4, p. 453–466.

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