# k-entry DFA

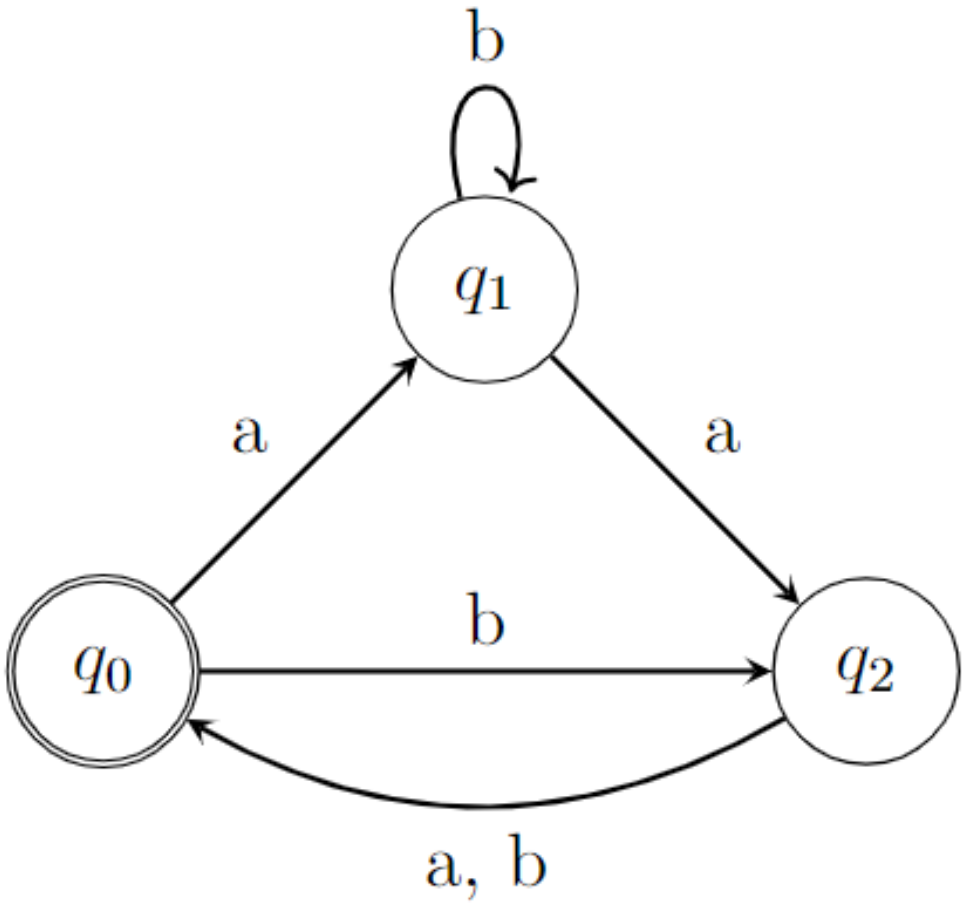
Šimon Huraj

# k-entry DFA

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

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

# k-entry DFA



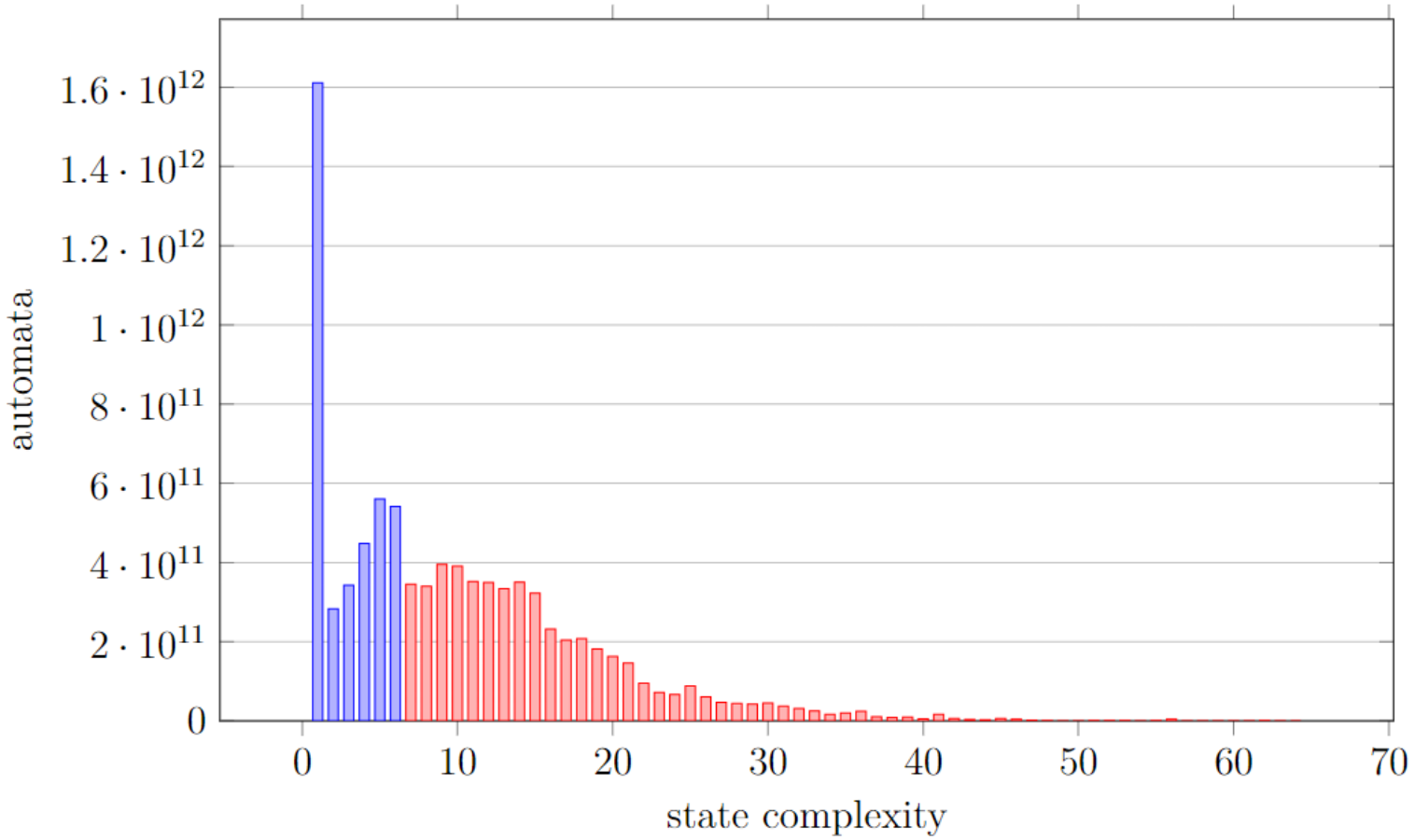
# Goals

- Develop a program that accepts an automaton as input and generates all automata with various choices of initial states. Expand this program to generate all n-state automata. Additionally, create a program capable of determinizing and minimizing the automaton to ascertain the state complexity of the language it represents. Furthermore, ensure that the program is designed to leverage parallel computing.
- Investigate the deterministic state complexity of automata represented by nondeterministic automata, where the only nondeterminism is from a choice of initial states.
- Examine the worst-case state complexity identified in 2.
- Explore the range of all obtainable state complexities from 2.
- Study the average state complexity from 2.

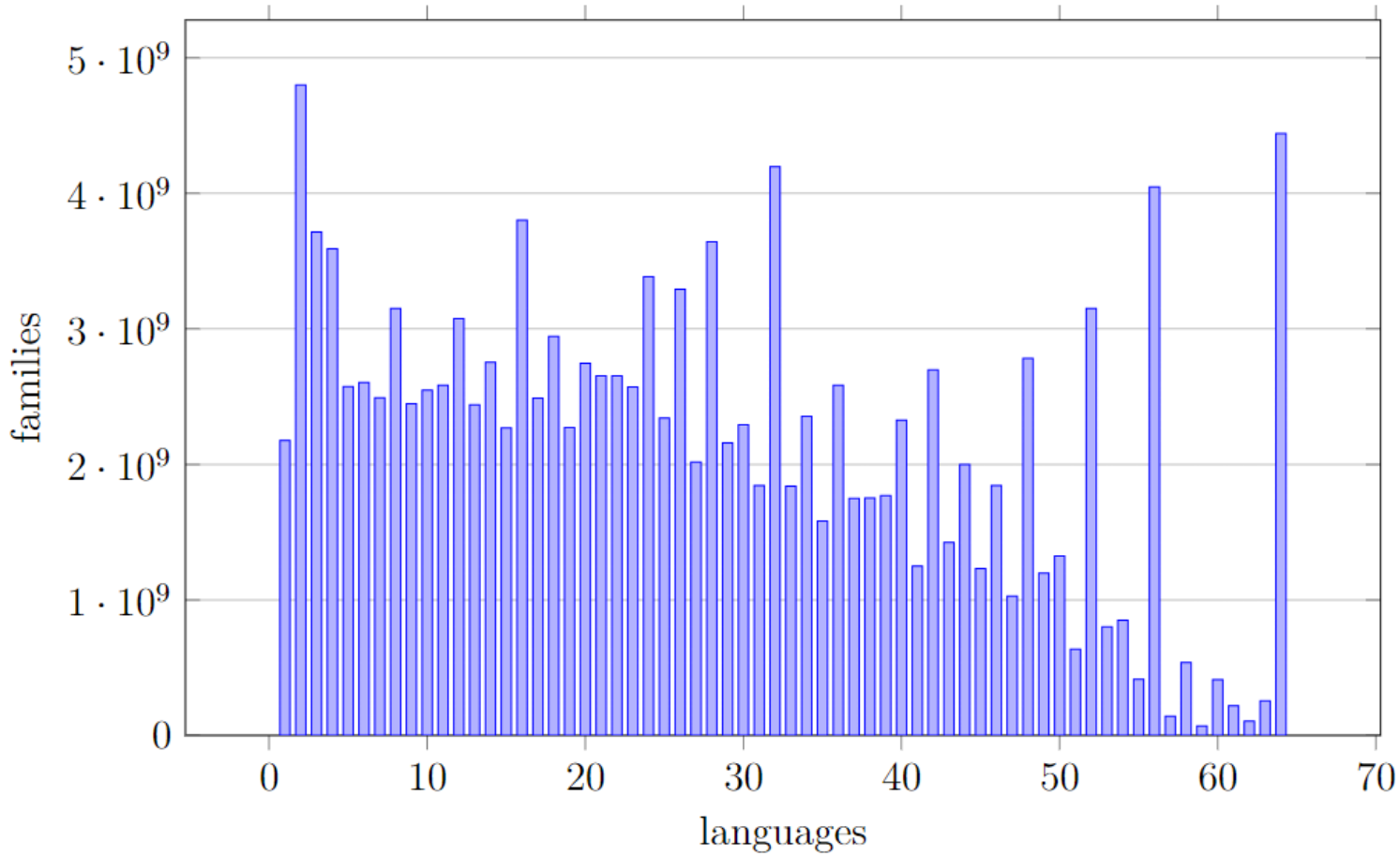
# Program

- Show C++

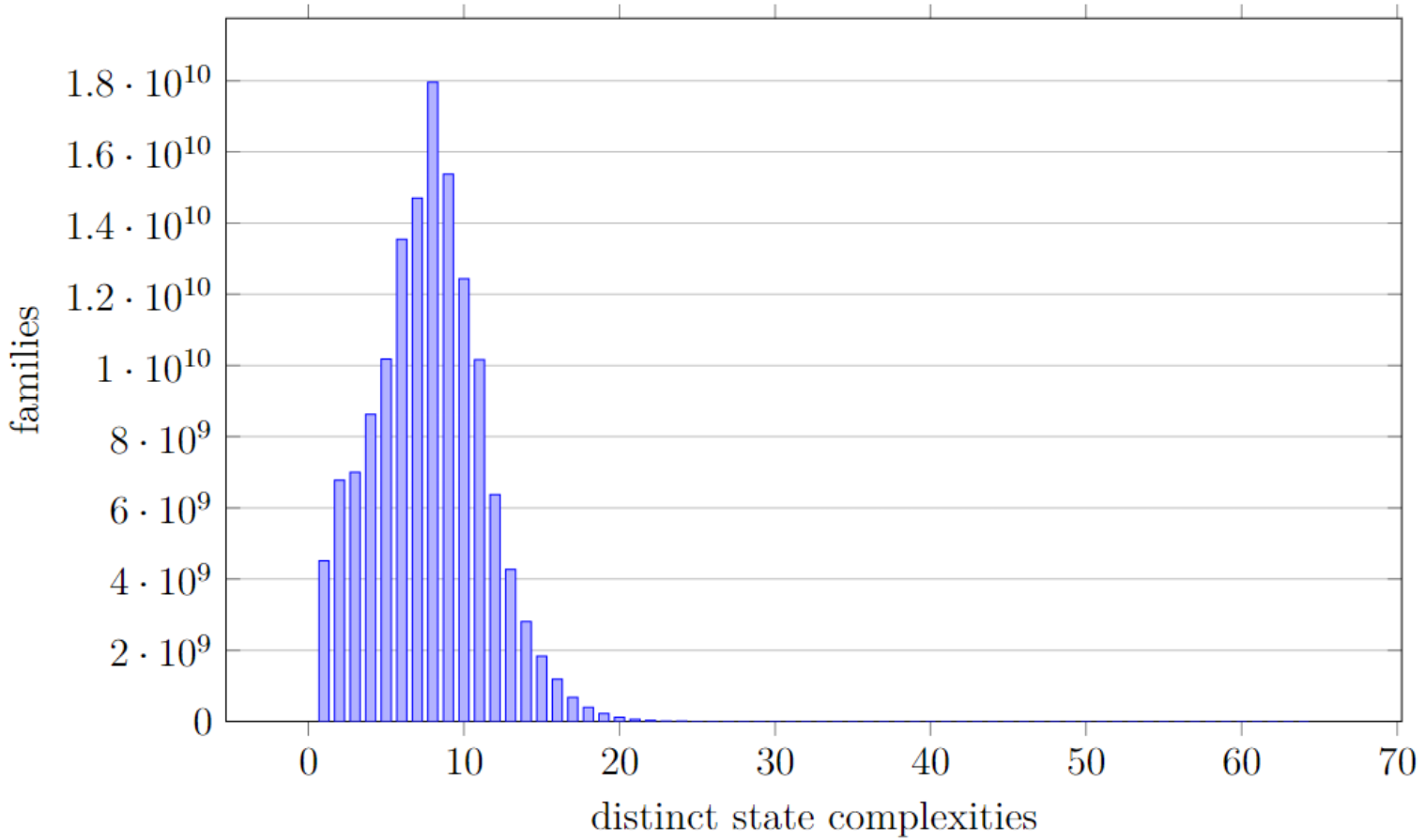
# Generation


automata
$1.6 \cdot 10^{12}$
$1.4 \cdot 10^{12}$
$1.2 \cdot 10^{12}$
$1 \cdot 10^{12}$
$8 \cdot 10^{11}$
$6 \cdot 10^{11}$
$4 \cdot 10^{11}$
$2 \cdot 10^{11}$
0
0
10
20
30
40
50
60
70
state complexity

# Generation


families
languages
0
1 · 10^9
2 · 10^9
3 · 10^9
4 · 10^9
5 · 10^9
0
10
20
30
40
50
60
70

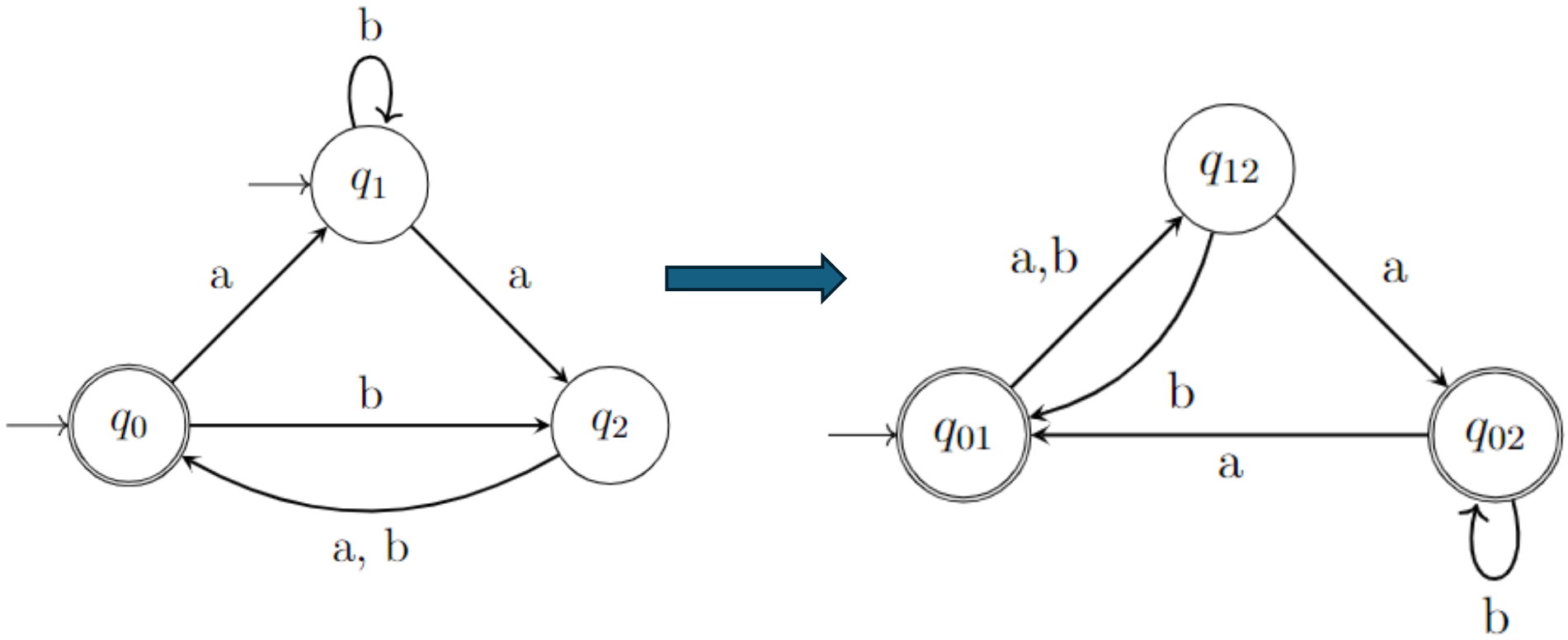
# Generation


families
$1.8 \cdot 10^{10}$
$1.6 \cdot 10^{10}$
$1.4 \cdot 10^{10}$
$1.2 \cdot 10^{10}$
$1 \cdot 10^{10}$
$8 \cdot 10^{9}$
$6 \cdot 10^{9}$
$4 \cdot 10^{9}$
$2 \cdot 10^{9}$
0
0
10
20
30
40
50
60
70
distinct state complexities

# Average State Complexity

- show theorem and proof

# Magic numbers



# Worst-case state complexity

- show lemma with proof