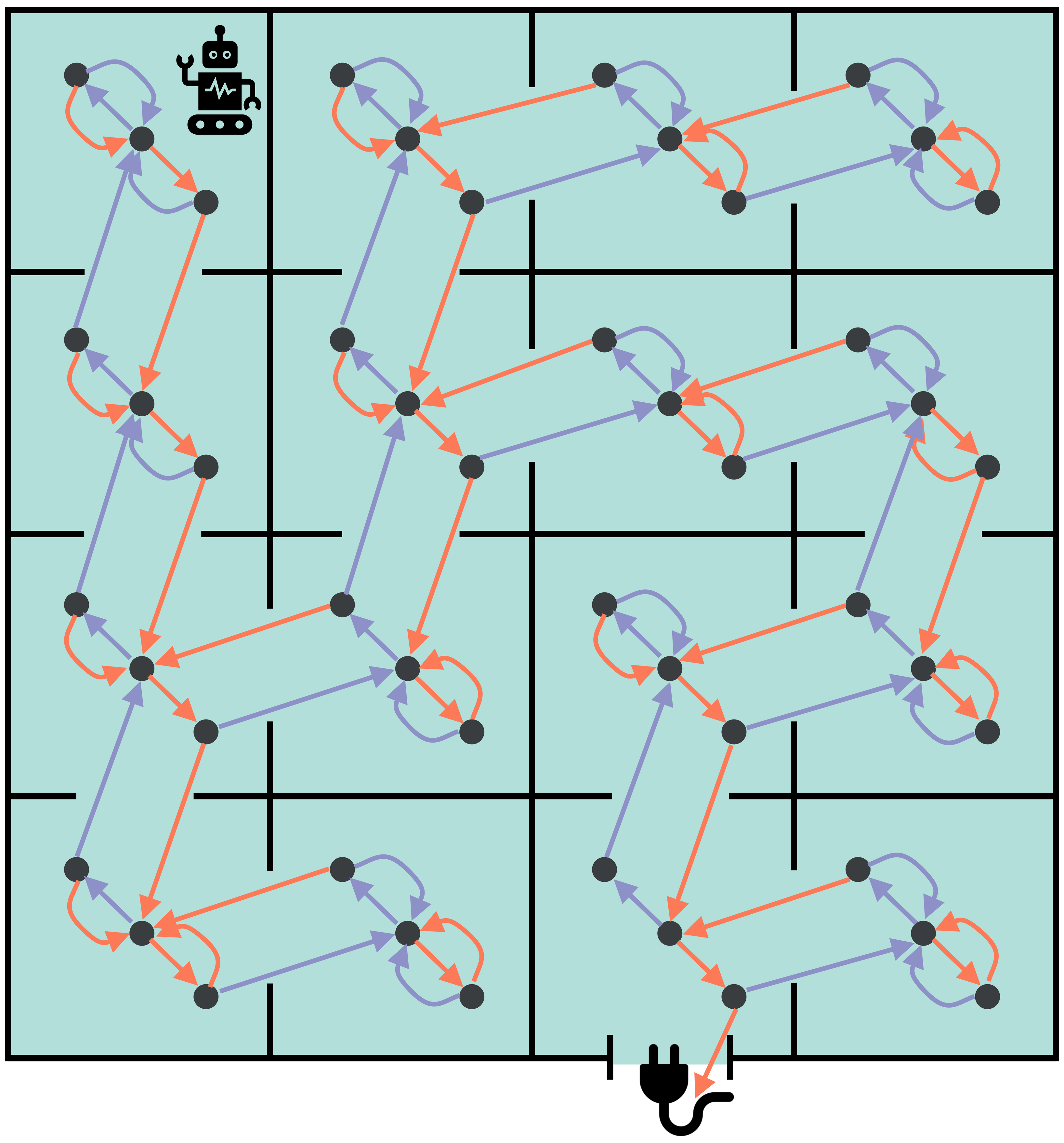
Study of Random w-trees and Automaton Synchronization

Attila Genda[1], **Guillem Perarnau**[2]

[1]Institut of Engineering Mechanics (ITM, KIT),

[2]School of Mathematics and Statistics (FME, UPC)



Guide the Robot Through the Labyrinth!

Game Rules

1. You can only step on the disks (●).
2. You always have two choices: Orange or Purple.
3. Limited Memory: max. 16 bits (e.g. OOPO = 4 bits).
4. Repeat the chosen sequence w until necessary.
5. You will start from a random position.

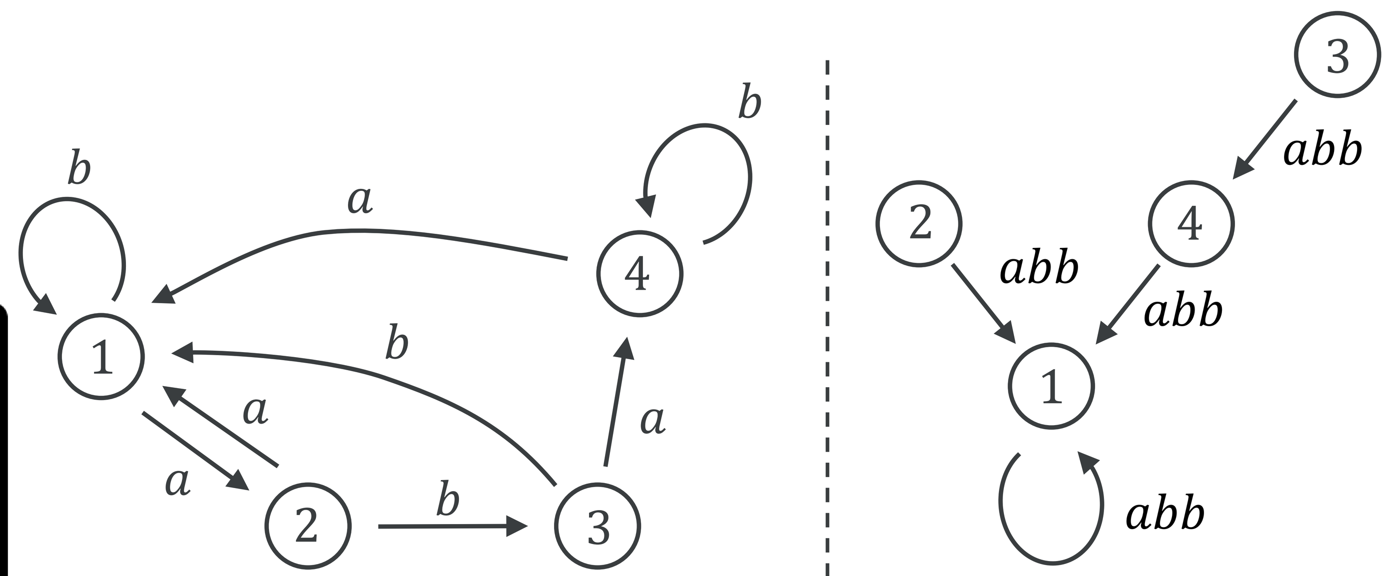
Find a synchronizing word ω of the form $\omega = w|w|\ldots|w$.

- What is the least necessary memory?
- How long will it take to escape the maze?
- How is escape time related to maze size n?

A „Lazy" Way to Synchronize: w-trees

Problem setting: random 2-letter DFA of size n

Idea: apply the same word w_k of length k repeatedly



SCAN ME

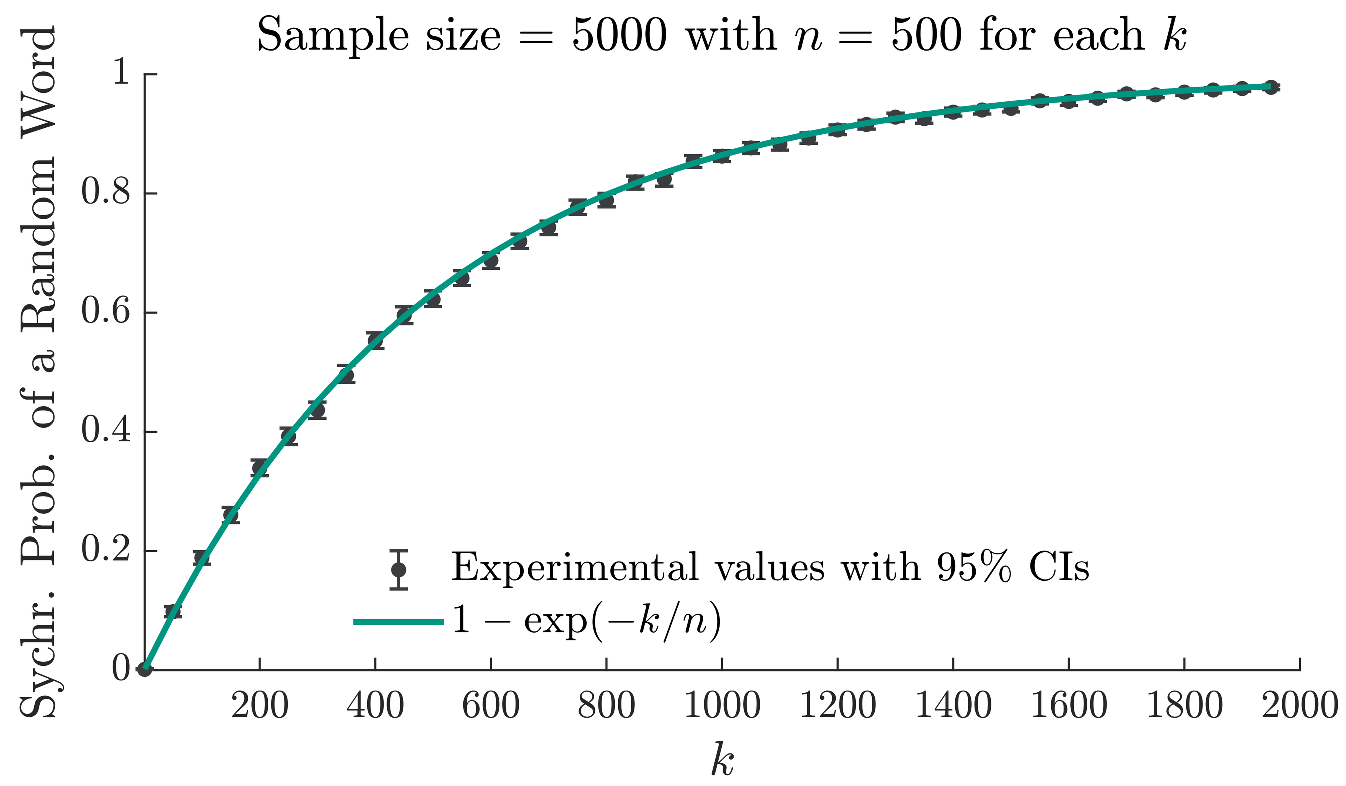
Best solution:

$7 \times$ OOOOOPPOPPPOP

Chances of Synchronization

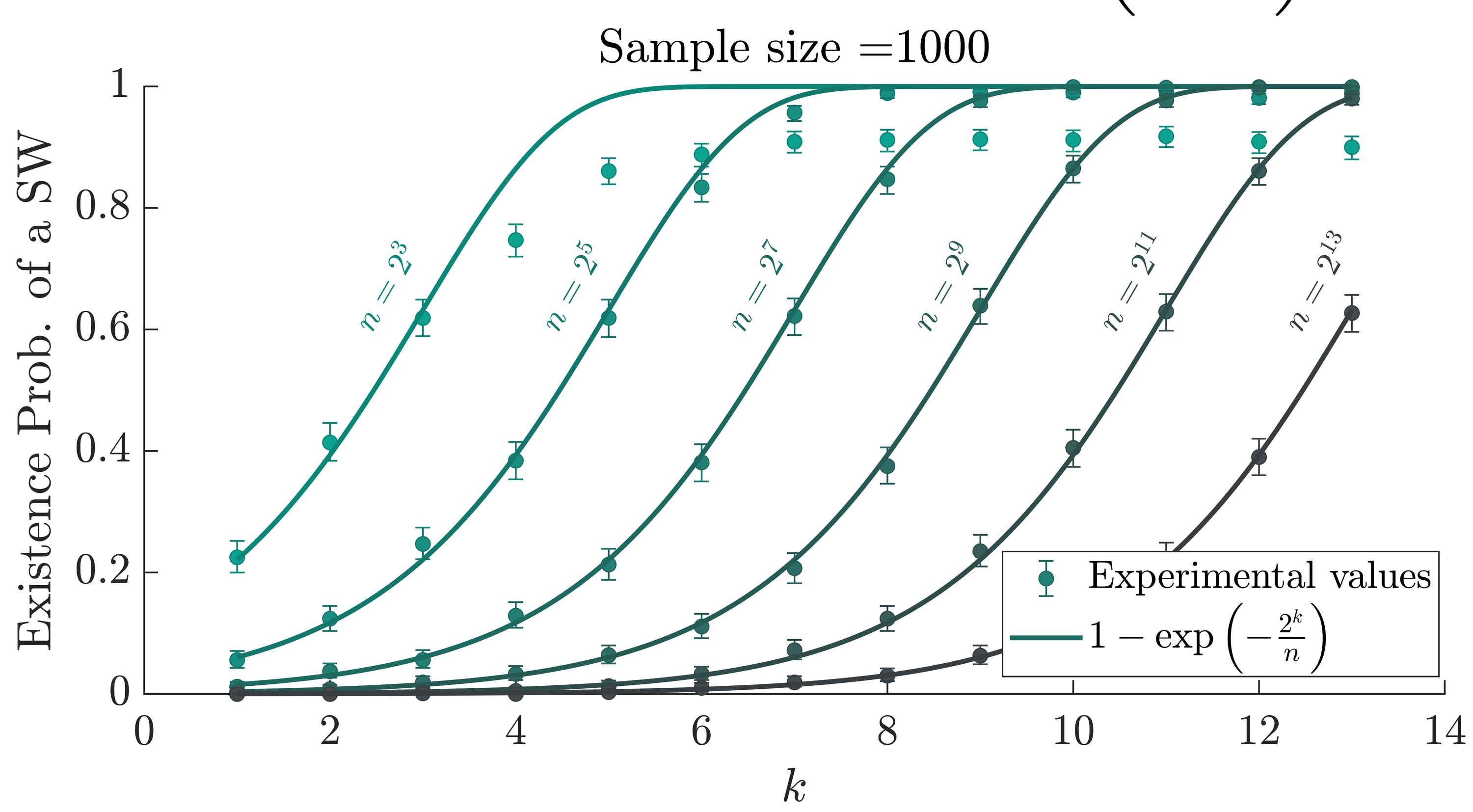
A repeated random word w_k of length k synchronizes a 2-letter random DFA A with n states with probability

$$P(\omega = w_k|w_k|\ldots|w_k \text{ synchs. } A) \approx 1 - \exp\left(-\frac{k}{n}\right).$$



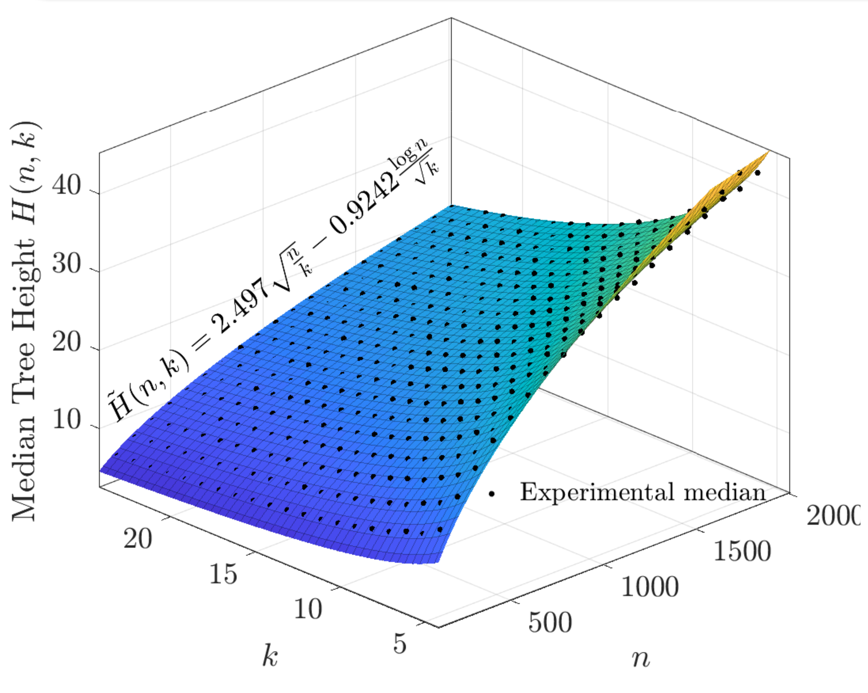
The probability that there is a word w_k of length k that synchronizes a random 2-letter DFA A for large n is

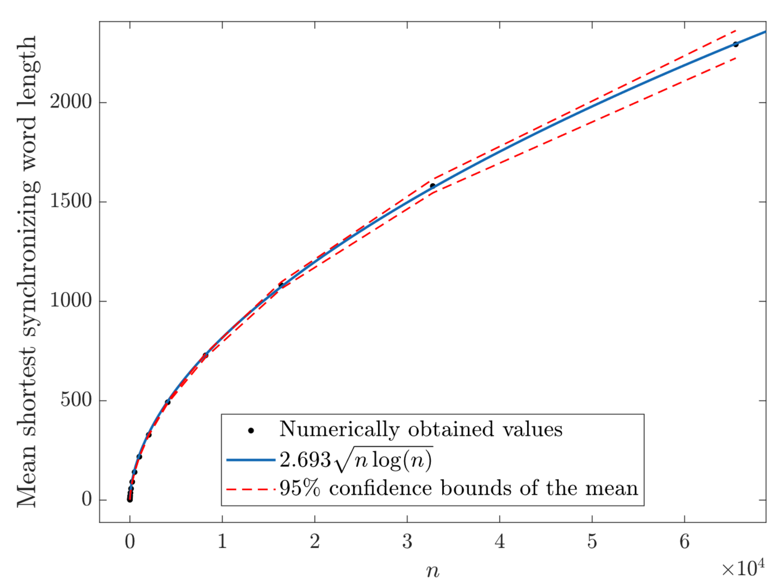
$$P(\exists w_k | w_k \text{ synchs. } A) \approx 1 - \exp\left(-\frac{2^k}{n}\right).$$



Consequence: to a random 2-letter DFA with high probability there is a synchronizing (SW) word of the form $\omega = w_k|\ldots|w_k$ where k is at most $\lceil \log_2 n + 1 \rceil$.

The Length of Syncronizing Words





Algorithm: finds short SWs for DFA with $n \approx 10^5$.

Mean SW length is $\approx 2.693\sqrt{n \log n}$.