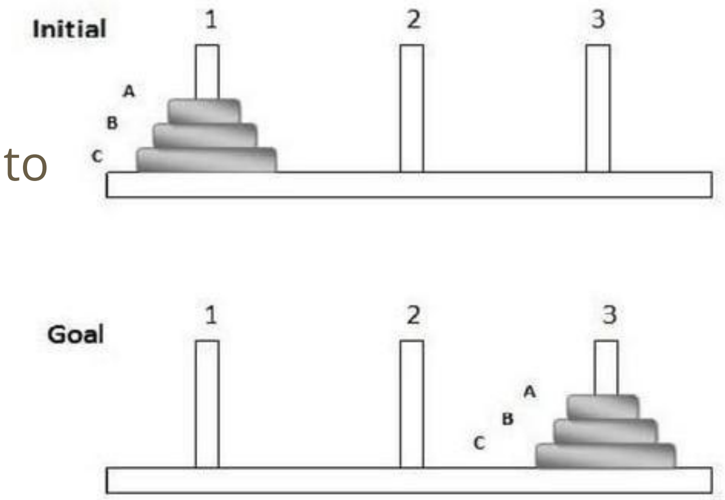
The word and order problems in the Hanoi Tower Groups

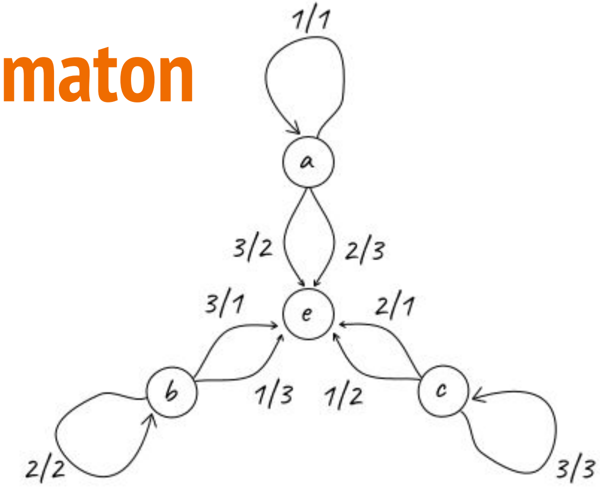
Davyd Zashkolnyi

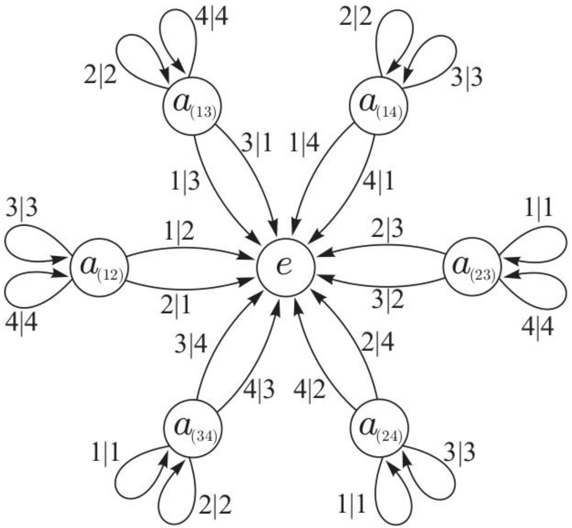
Hanoi Tower Game

- Classically is played on 3 pegs with n disks
- The goal is to move all disks from one peg to another one not placing larger disk on the smaller one
- Complexity - O(2^n)



Hanoi Automaton

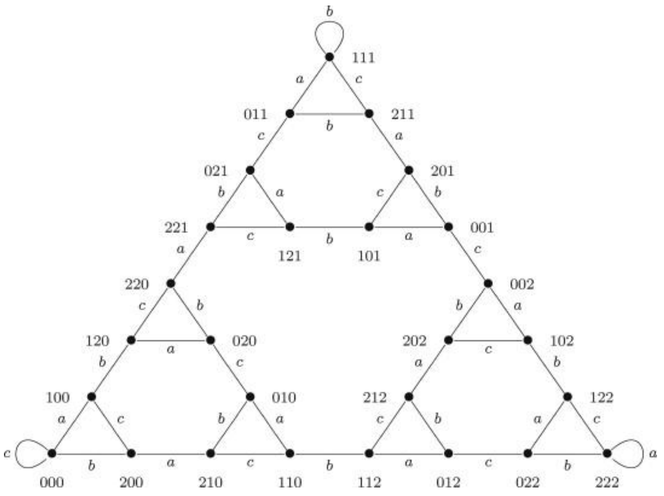


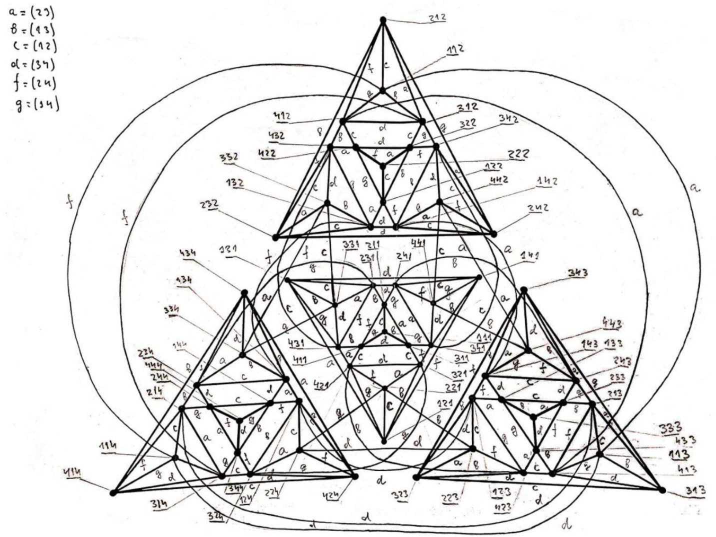


- Game on k pegs can be modelled by automaton handling words on alphabet {1, 2, 3, … k} (above automata for 3 and 4 pegs respectively)
- Each state of the automaton is also recursive function

$$a_{(ij)}(iv) = jv, \quad a_{(ij)}(jv) = iv, \quad a_{(ij)}(xv) = xa_{(ij)}(v) \ \text{ for } x \notin \{i, j\}.$$

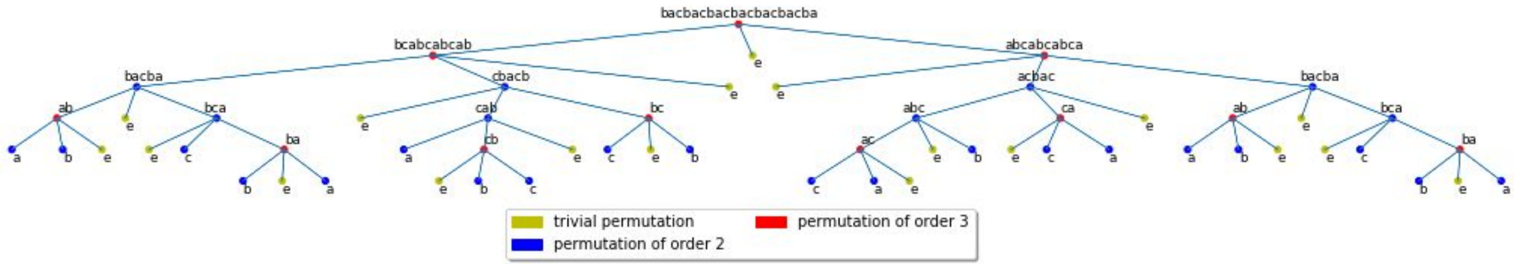
Hanoi Tower Groups





- Aforementioned recursions form a set of finite-generated self-similar groups, a.k.a. Hanoi Tower Groups on k pegs H_k, which also are examples of so-called Automaton groups
- Schreier graph of action H_k on n-length words of {1, ... k}* simulates Hanoi Tower Game on k pegs with n disks

Word problem. Main results

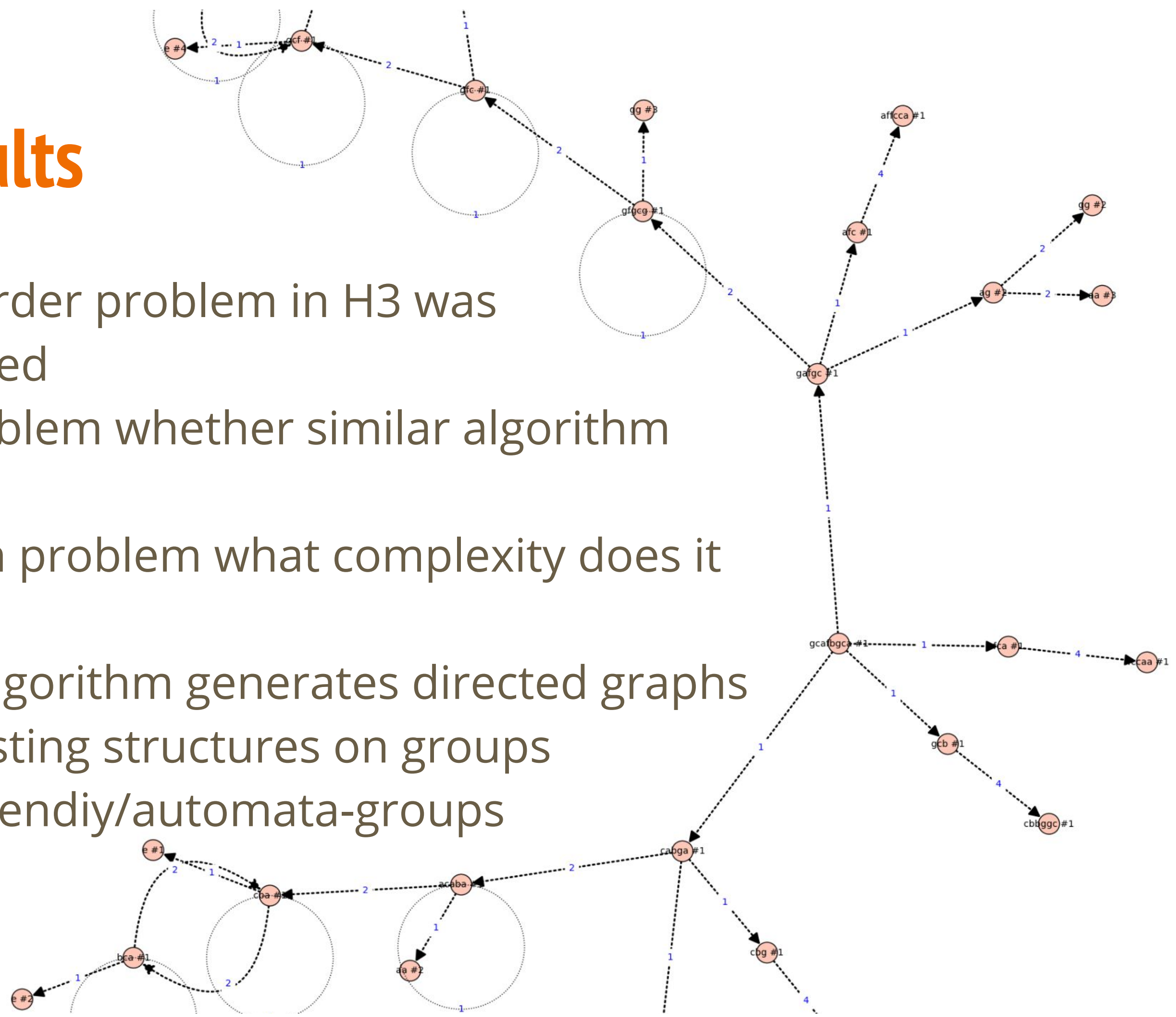


- Algorithm for solving word problem has complexity O(size(w))
- Elements like (abc)^k up to renaming have the maximum possible size a(n), where n is length
- **Theorem 2.** *Exact form of function $a(n)$:*

$$a(n) = \begin{cases} n\left(\lfloor \log_2 n + 1 \rfloor\right) + 2n - 2^{\lfloor \log_2 n \rfloor + 1}, & n > 0 \\ a(0) = 0 \end{cases} \tag{3}$$

Order problem. Results

- Algorithm for solving order problem in H3 was proved and implemented
- It remains an open problem whether similar algorithm always stops in H4
- It also remains an open problem what complexity does it have in H3
- During execution the algorithm generates directed graphs which introduce interesting structures on groups
- https://github.com/davendiy/automata-groups



Thank you for attention.