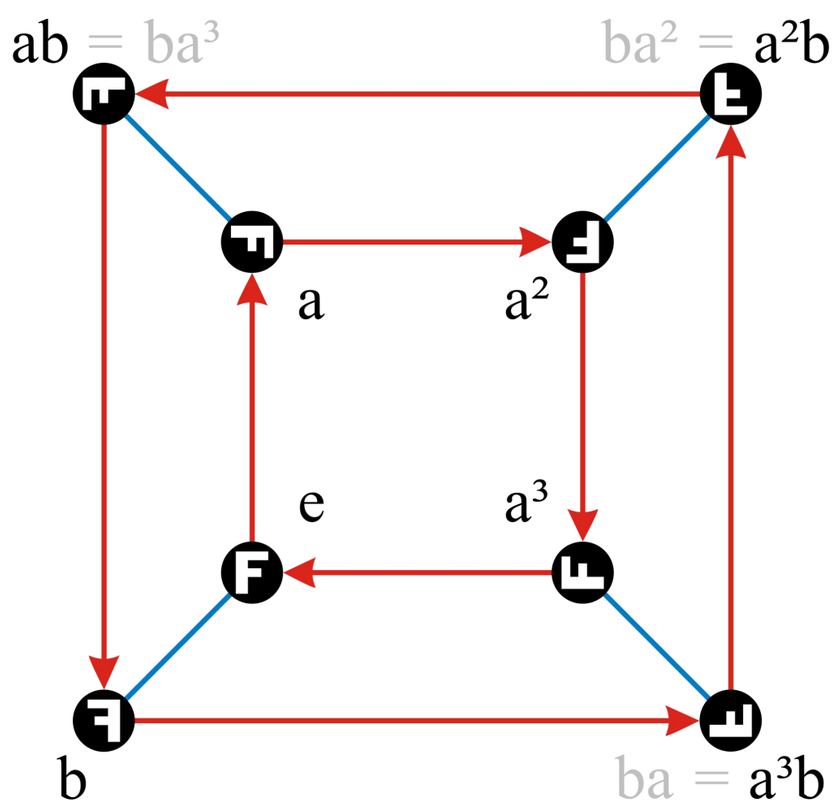
Cayley Maze: Reinforcement Learning environment with variable complexity

Artsiom Ranchynski, Łukasz Kuciński

Cayley Maze

Cayley Maze is an open-ended Reinforcement Learning environment, where the agent moves along the graph edges in order to reach exit vertex. If the underlying graph is a Cayley Graph - particular construction, preserving group symmetries, we call it Natural Cayley Maze. Rubik's Cube and array sorting problems are examples of Natural Cayley Mazes.



$$a = \begin{pmatrix} e & a & a^2 & a^3 & b & ab & a^2b & a^3b \\ a & a^2 & a^3 & e & a^3b & b & ab & a^2b \end{pmatrix},$$

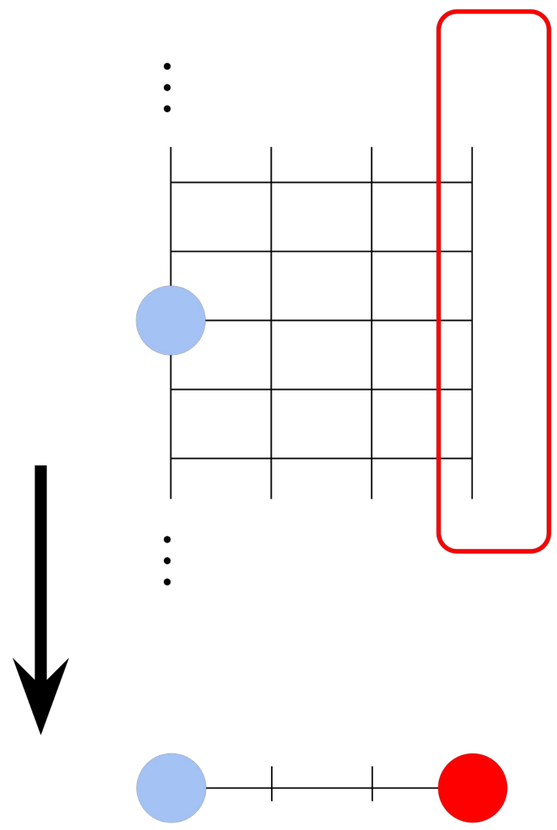
$$b = \begin{pmatrix} e & a & a^2 & a^3 & b & ab & a^2b & a^3b \\ b & ab & a^2b & a^3b & e & a & a^2 & a^3 \end{pmatrix}$$

Properties of Cayley Maze

- **Cayley Maze is universal**: every deterministic sparse MDP is an instance of Cayley Maze
- **Cayley Maze has variable computational complexity**: sorting is polynomial, while finding the optimal solution of Rubik's Cube is NP-complete
- **Cayley Maze allows to modify and simplify environments**
- **Cayley Maze allows to control the size of meta action space**: the action space of environment instances construction
- **Local properties of Natural Cayley Mazes become global:** if moving twice right is equal to moving left at one state, then it's true for all states

Complexity of MDP

Every sparse deterministic MDP induces Deterministic State Automata in an obvious way, and vice versa.
We define the **complexity of MDP** as the size of the state space of its minimal automaton, which exist by Myhill-Nerode theorem.
Actually, every deterministic MDP induces a transducer (string-to-string automaton) which also can be minimized, hence the definition can be generalized to non-sparse rewards.



Applications

- Since Cayley Maze allows to sample any MDP, it may be used for the evaluation of Unsupervised Environment Design algorithms in the classical scenario
- The symmetry of Natural Cayley Mazes allows to efficiently evaluate agent's generalization capabilities: for example, agent can be trained on the certain part of graph, and evaluated on unseen part
- It is possible not only to evaluate on subfamilies of environments, but on its various constructions. For example, we can check, whether the agent, which performed well on some instances would perform well on its product.
- Cayley Maze allows to construct environment curriculum with the guaranteed lower bound complexity, for example by sample Cayley Graphs of the simple groups (hence it's corresponding automaton can't be reduced)