Model learning in strongly structured environments[1]

Gábor Bartók, Csaba Szepesvári, Sandra Zilles

Department of Computing Science
University of Alberta

Barbados 16/04/2009

[1]Based on the paper G. Bartók, Cs. Szepesvári and S. Zilles: *Active Learning of Group-Structured Environments.* ALT '08

Outline

Outline

1. ~~Motivation~~
2. Different representations

Outline

1. ~~Motivation~~
2. Different representations
3. The learning model

Outline

1. ~~Motivation~~
2. Different representations
3. The learning model
4. A not-so-promising theorem

Outline

1. ~~Motivation~~
2. Different representations
3. The learning model
4. A not-so-promising theorem
5. Examples of learnable models

- ~~Reinforcement Learning solution vs. Intuition~~

- ~~Reinforcement Learning solution vs. Intuition~~

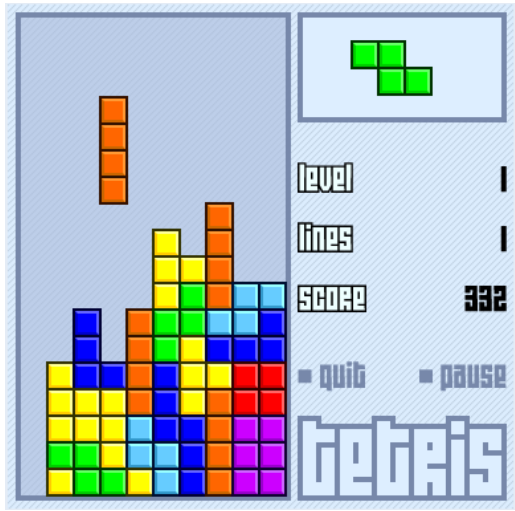
  ~~Examples:~~

  - ~~*Learning to Acquire the Ball* (Fidelman, Stone)~~

- ~~Reinforcement Learning solution vs. Intuition~~

  ~~Examples:~~

  - ~~*Learning to Acquire the Ball* (Fidelman, Stone)~~
  - ~~Tetris~~



- ~~Reinforcement Learning solution vs. Intuition~~

  ~~Examples:~~

  - ~~*Learning to Acquire the Ball* (Fidelman, Stone)~~
  - ~~Tetris~~
- ~~Best model representation vs. Intuition?~~

- ~~Reinforcement Learning solution vs. Intuition~~

  ~~Examples:~~

  - ~~*Learning to Acquire the Ball* (Fidelman, Stone)~~
  - ~~Tetris~~
- ~~Best model representation vs. Intuition?~~
- ~~The choice of the representation can make a difference~~

- ~~Reinforcement Learning solution vs. Intuition~~

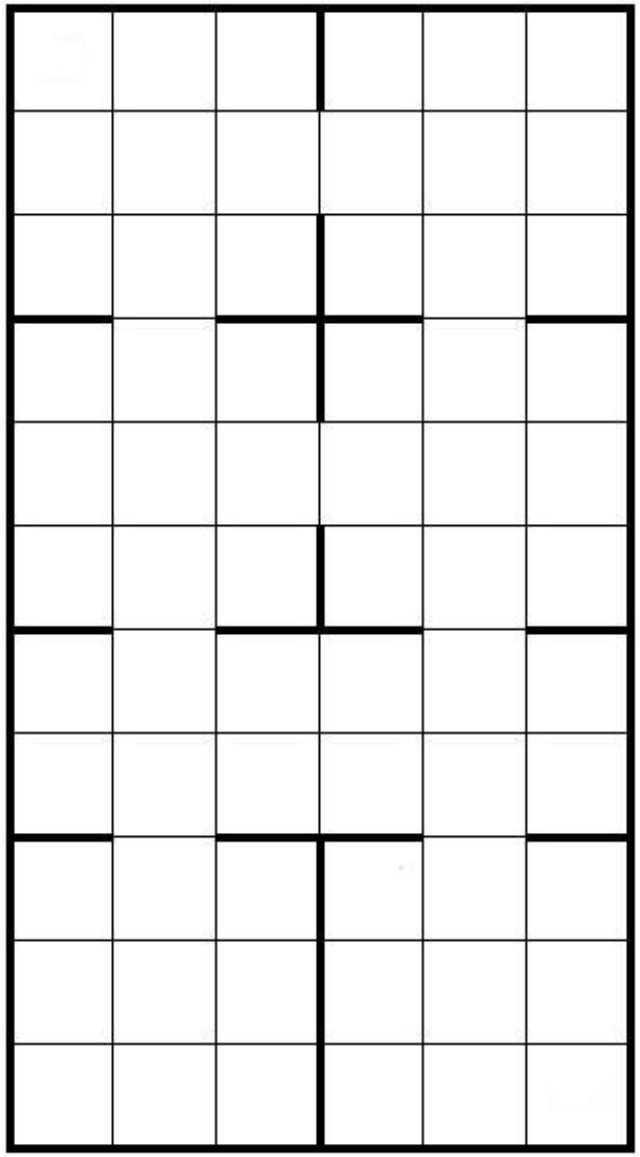
  ~~Examples:~~

  - ~~*Learning to Acquire the Ball* (Fidelman, Stone)~~
  - ~~Tetris~~
- ~~Best model representation vs. Intuition?~~
- ~~The choice of the representation can make a difference~~
- ~~Representation: description of the model~~

Example 1

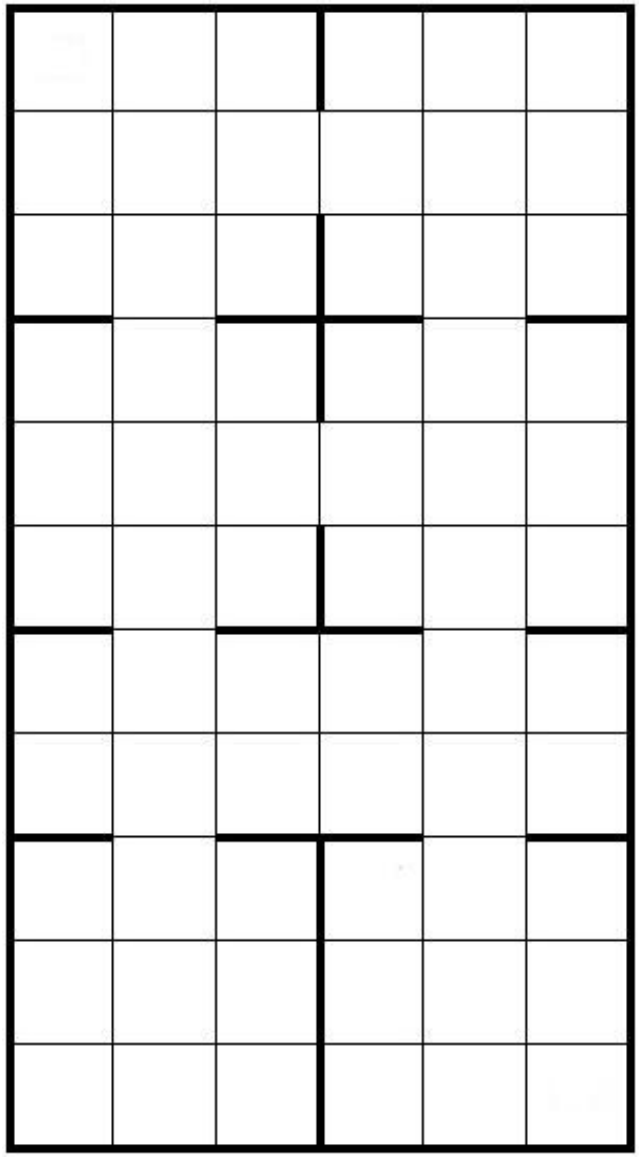
Example 1

- Two-dimensional gridworld



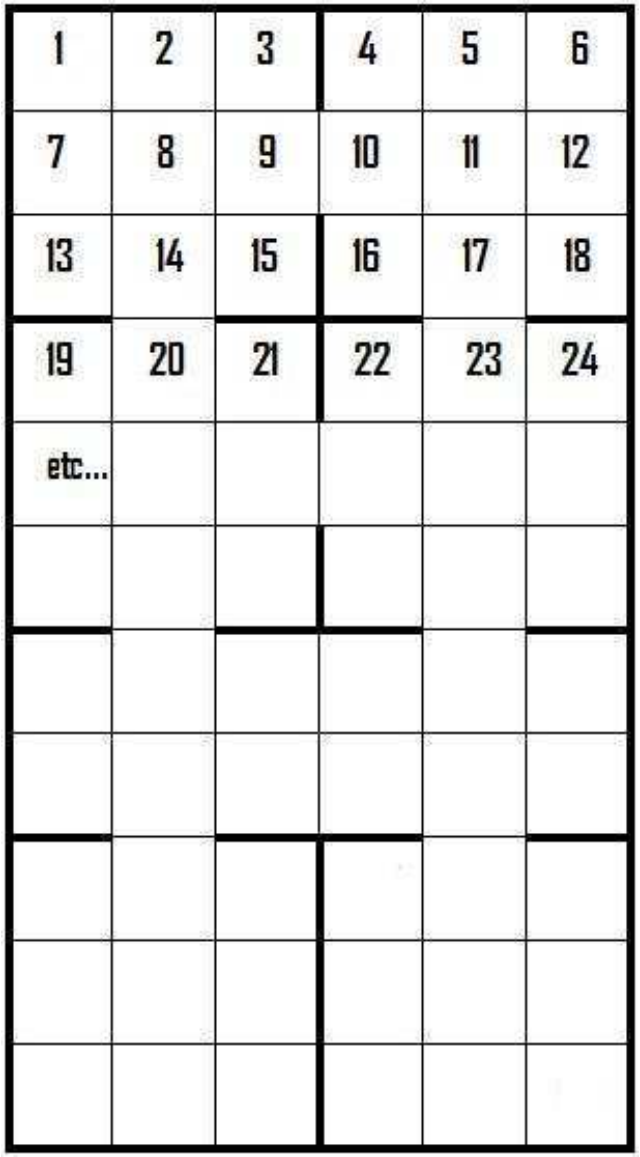
Example 1

- Two-dimensional gridworld
- Two possible representations:



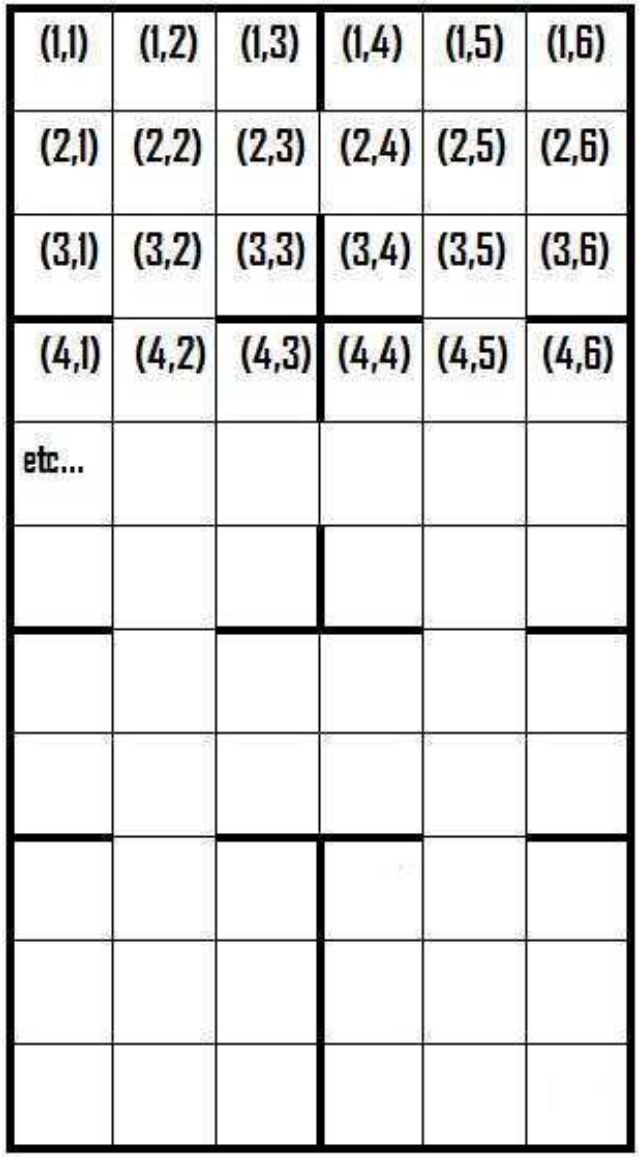
Example 1

- Two-dimensional gridworld
- Two possible representations:
  1. Numbering the states



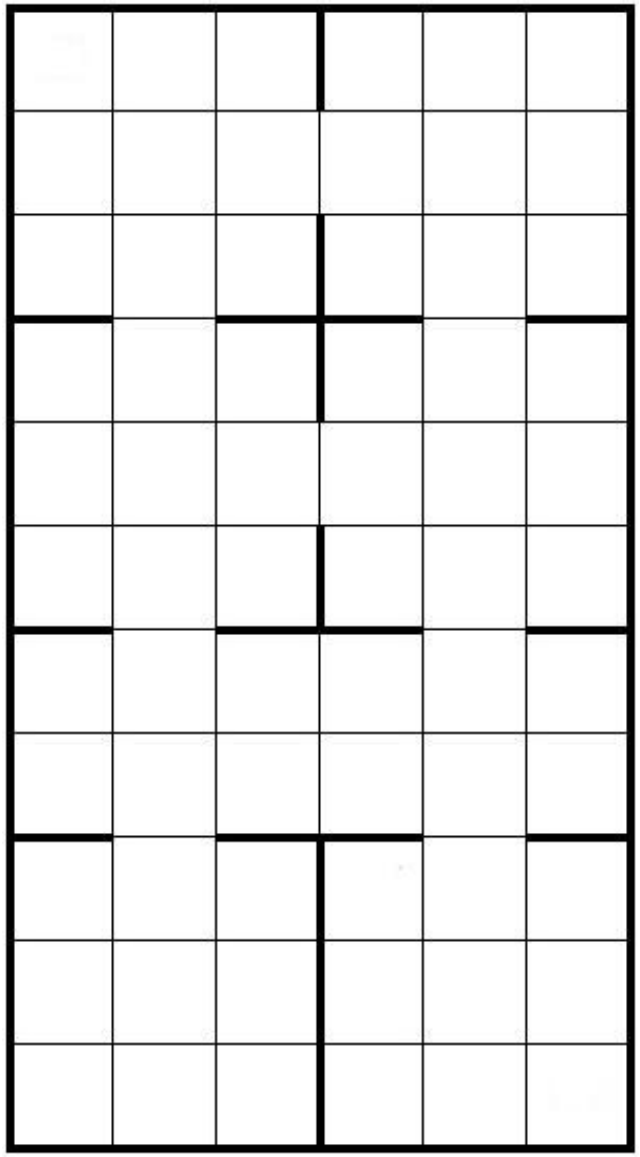
Example 1

- Two-dimensional gridworld
- Two possible representations:
  1. Numbering the states
  2. Coordinates (x, y)



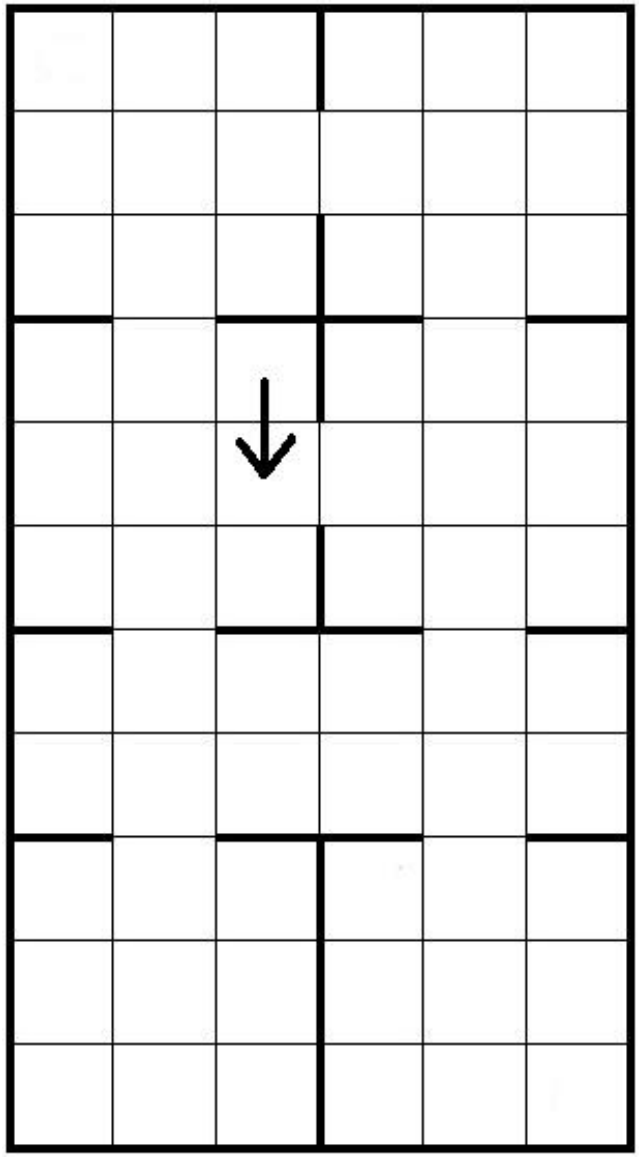
Example 1

- Two-dimensional gridworld
- Two possible representations:
  1. Numbering the states
  2. Coordinates (x, y)
- Which one is better?



Example 1

- Two-dimensional gridworld
- Two possible representations:
  1. Numbering the states
  2. Coordinates (x, y)
- Which one is better?
- Coordinates help to describe the actions



Example 2

Example 2

- Atari game

Example 2

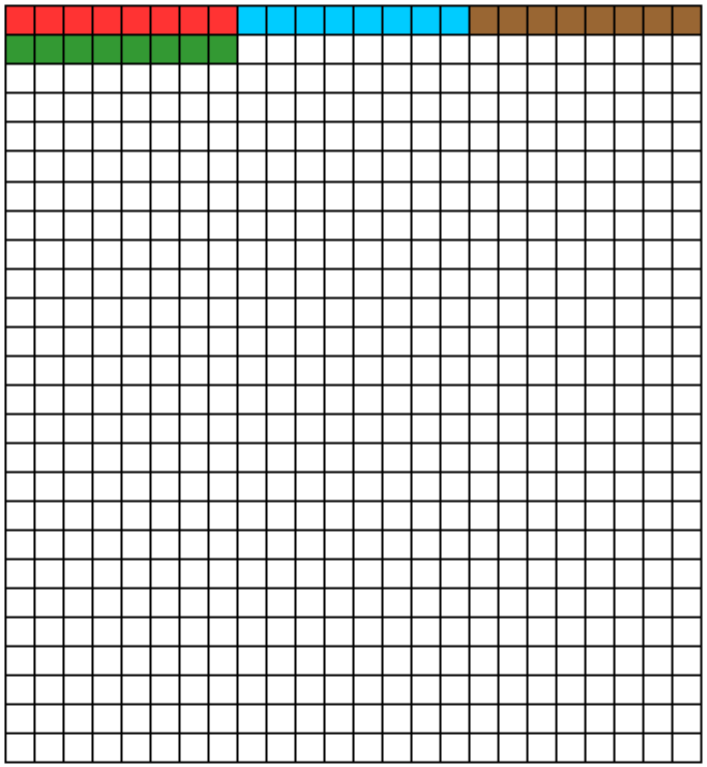
- Atari game
- 1024 bits of memory (internal state)

Example 2

- Atari game
- 1024 bits of memory (internal state)
- That's 2^{1024} states!!

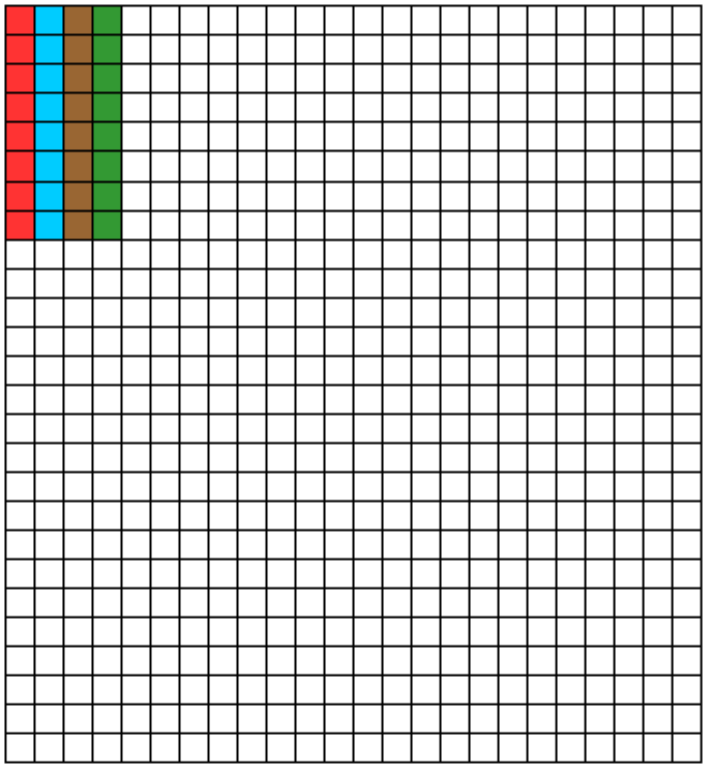
Example 2

- Atari game
- 1024 bits of memory (internal state)
- That's 2^{1024} states!!
- How are the bits grouped together? Like this?



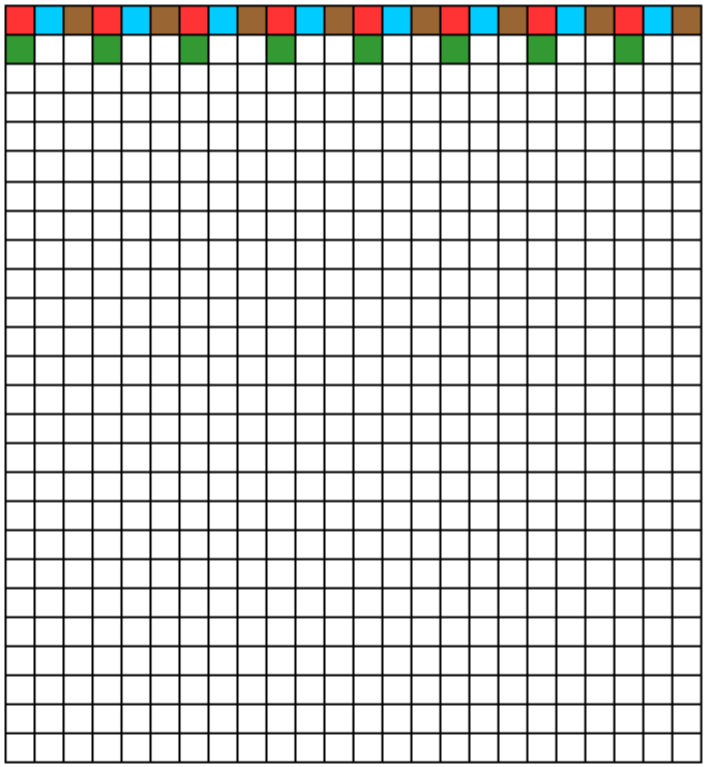
Example 2

- Atari game
- 1024 bits of memory (internal state)
- That's 2^{1024} states!!
- How are the bits grouped together? Like this?
- Or this?



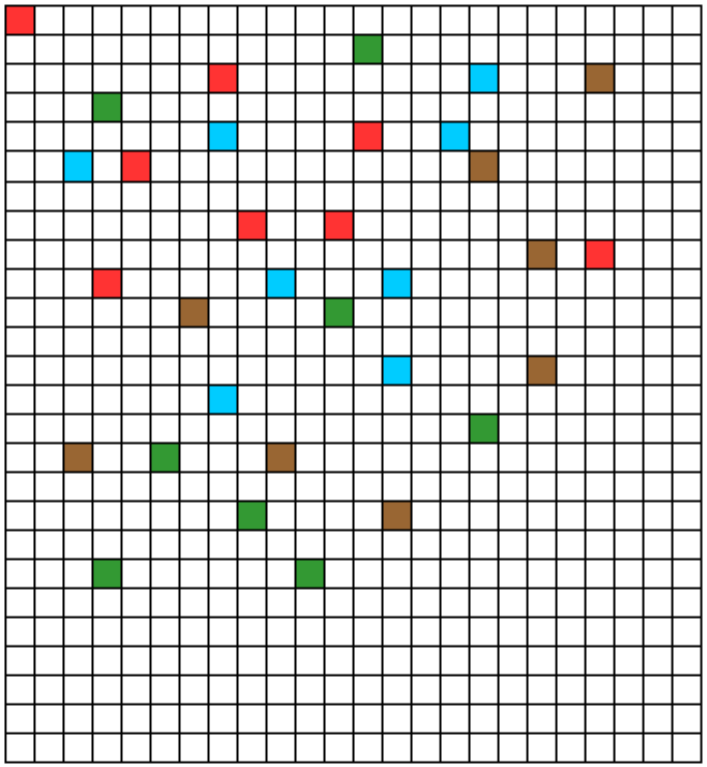
Example 2

- Atari game
- 1024 bits of memory (internal state)
- That's 2^{1024} states!!
- How are the bits grouped together? Like this?
- Or this?
- Ouch!



Example 2

- Atari game
- 1024 bits of memory (internal state)
- That's 2^{1024} states!!
- How are the bits grouped together? Like this?
- Or this?
- Ouch!
- Aaaaargh!



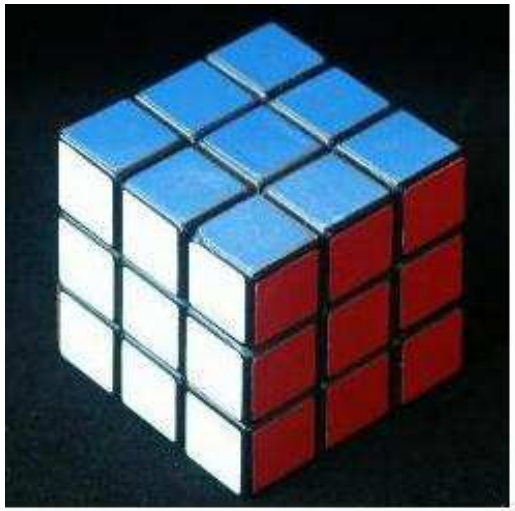
Example 3

Example 3

- Rubik's cube

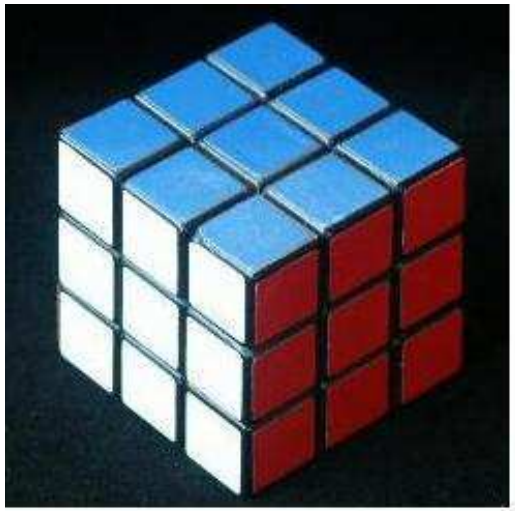
Example 3

- Rubik's cube
- Very strong structure



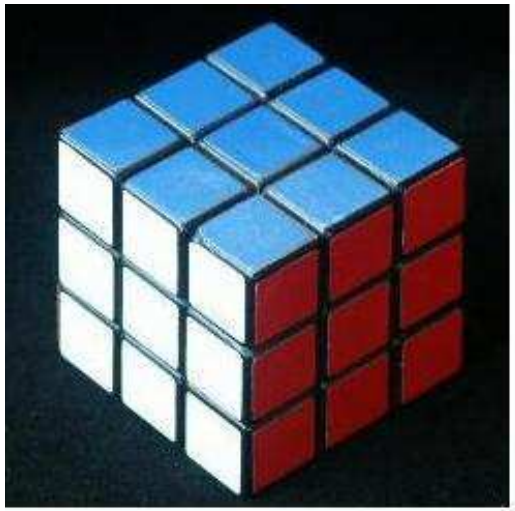
Example 3

- Rubik's cube
- Very strong structure
- The representation is what we see:



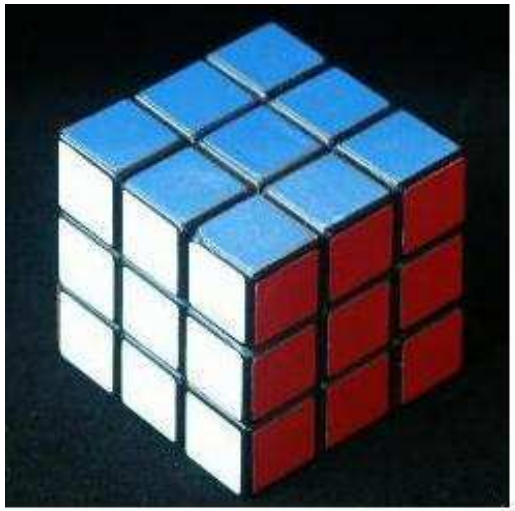
Example 3

- Rubik's cube
- Very strong structure
- The representation is what we see:
  - 3 dimensions



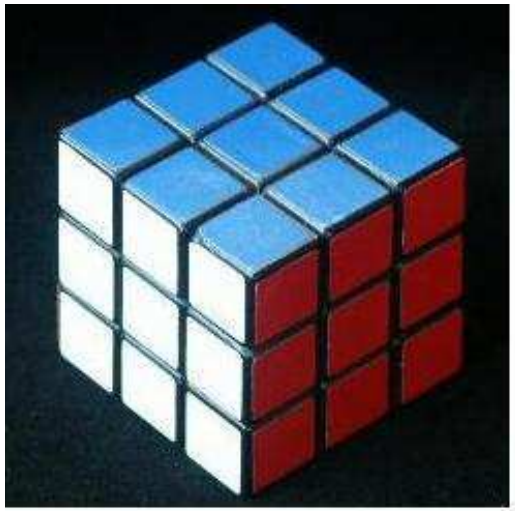
Example 3

- Rubik's cube
- Very strong structure
- The representation is what we see:
  - 3 dimensions
  - tiles



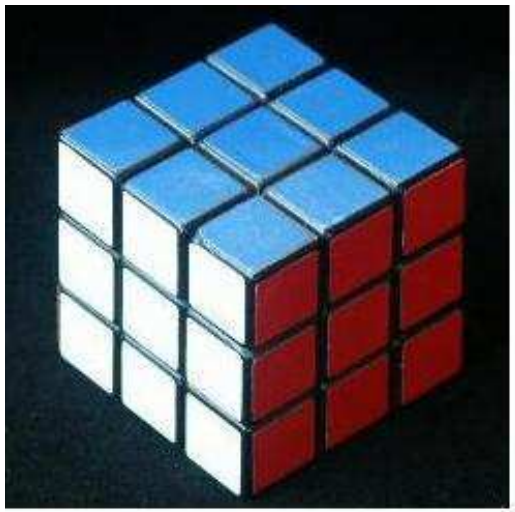
Example 3

- Rubik's cube
- Very strong structure
- The representation is what we see:
  - 3 dimensions
  - tiles
  - colours



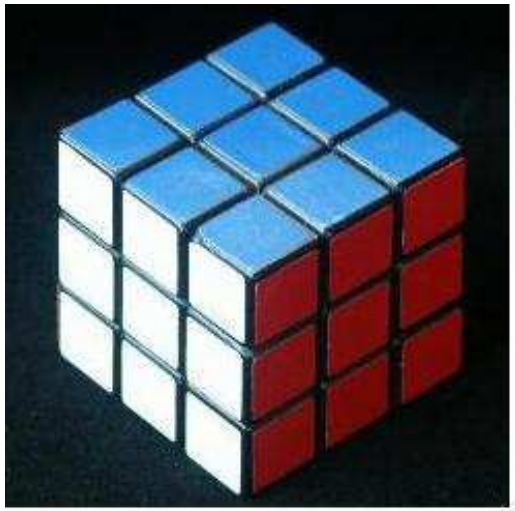
Example 3

- Rubik's cube
- Very strong structure
- The representation is what we see:
  - 3 dimensions
  - tiles
  - colours
  - actions: rotations



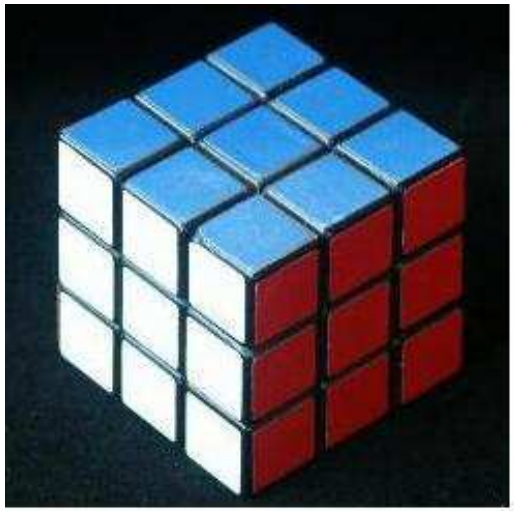
Example 3

- Rubik's cube
- Very strong structure
- The representation is what we see:
  - 3 dimensions
  - tiles
  - colours
  - actions: rotations
- Is that the best?

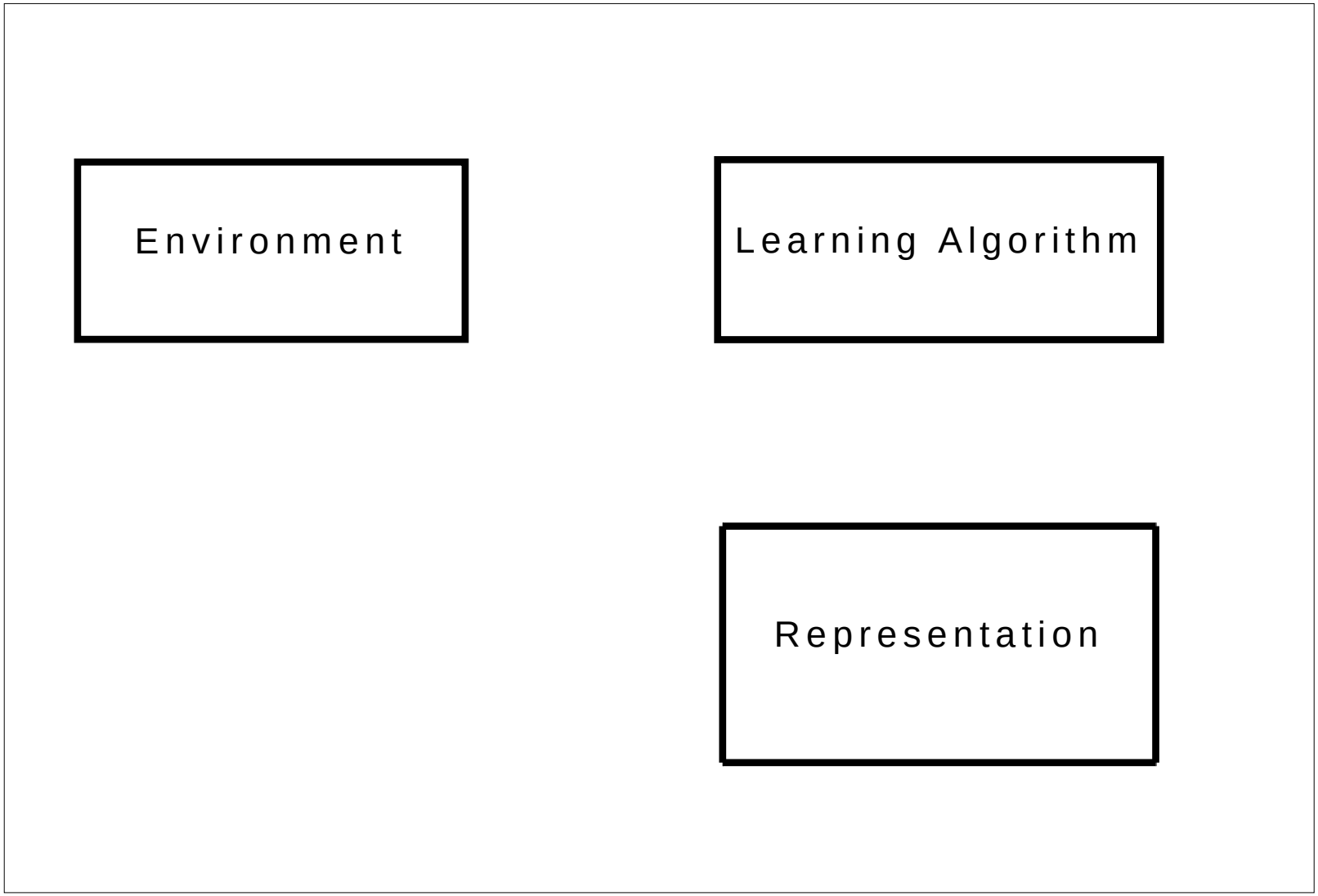


Example 3

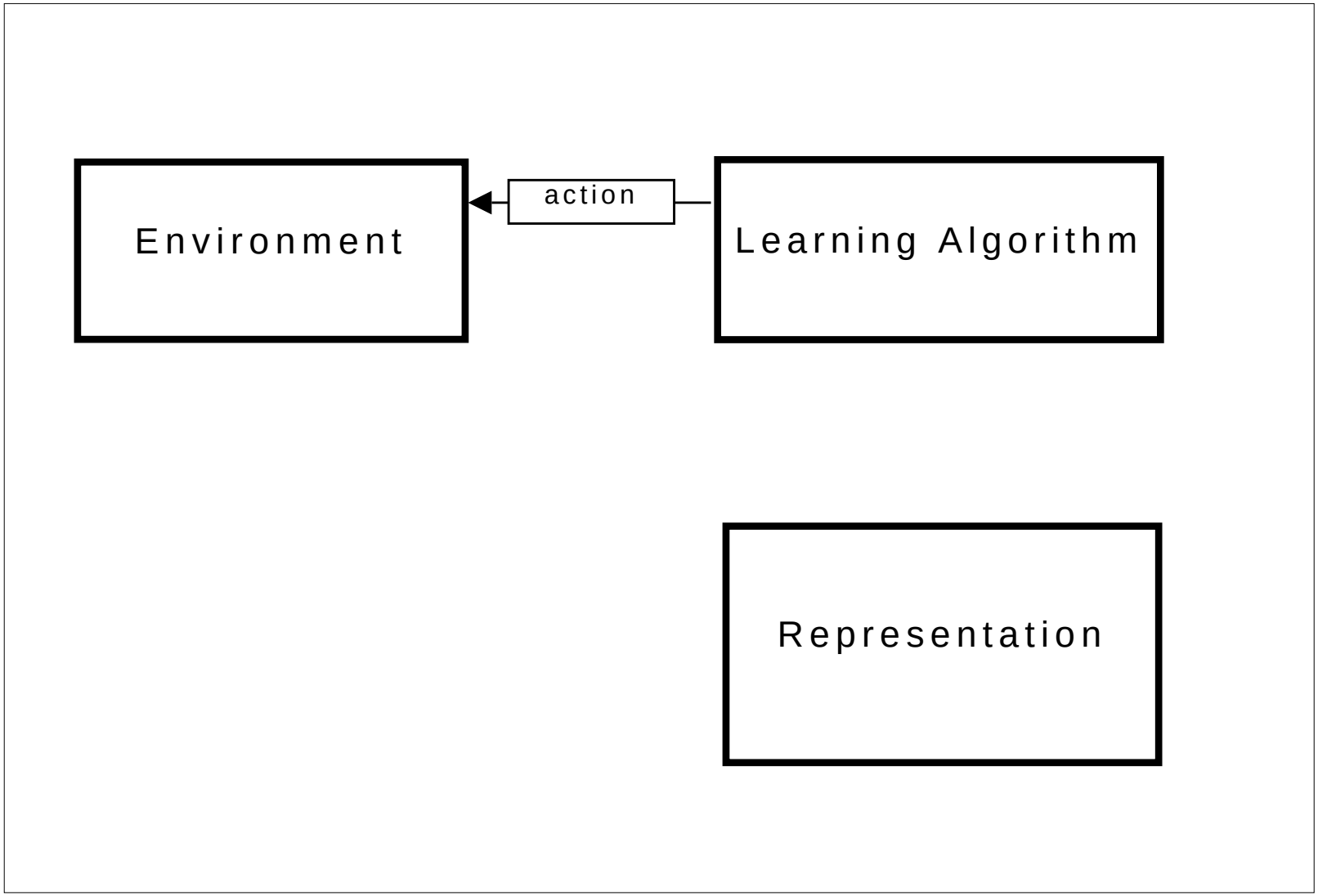
- Rubik's cube
- Very strong structure
- The representation is what we see:
  - 3 dimensions
  - tiles
  - colours
  - actions: rotations
- Is that the best?
- ...who knows



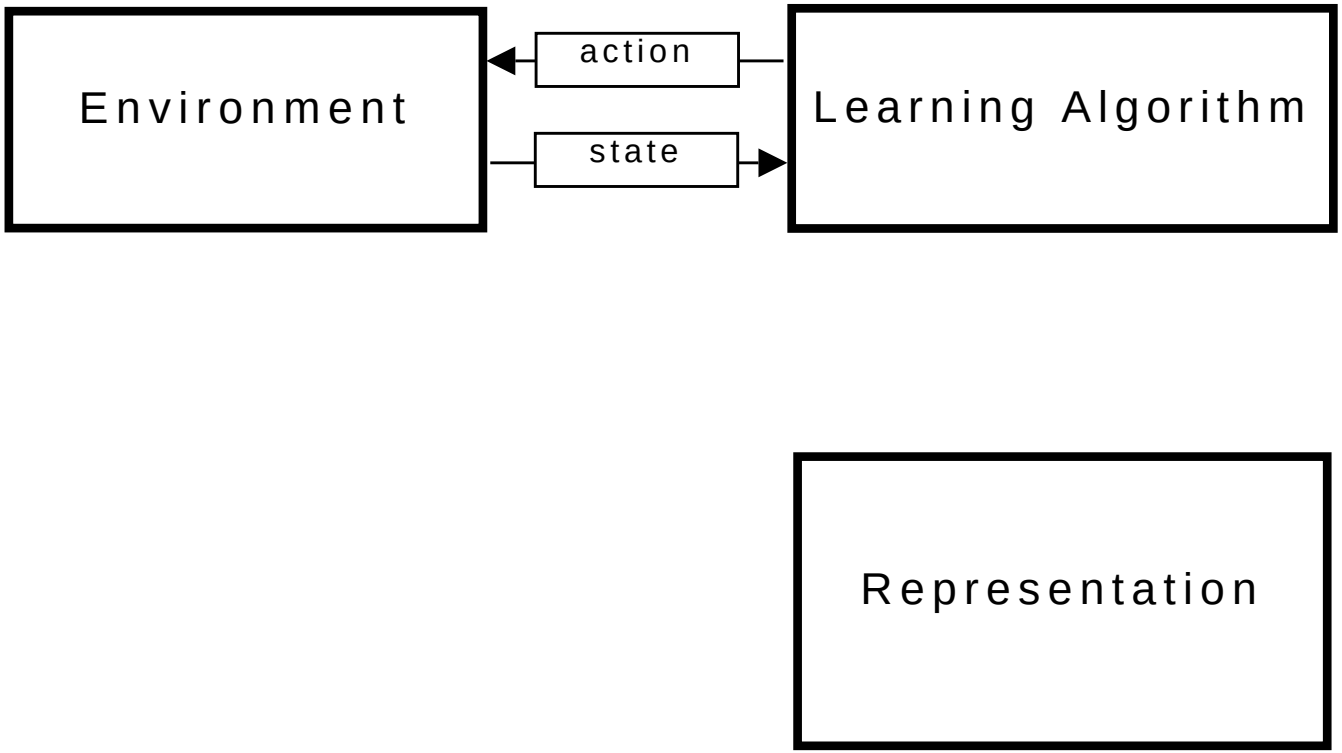
Our Model



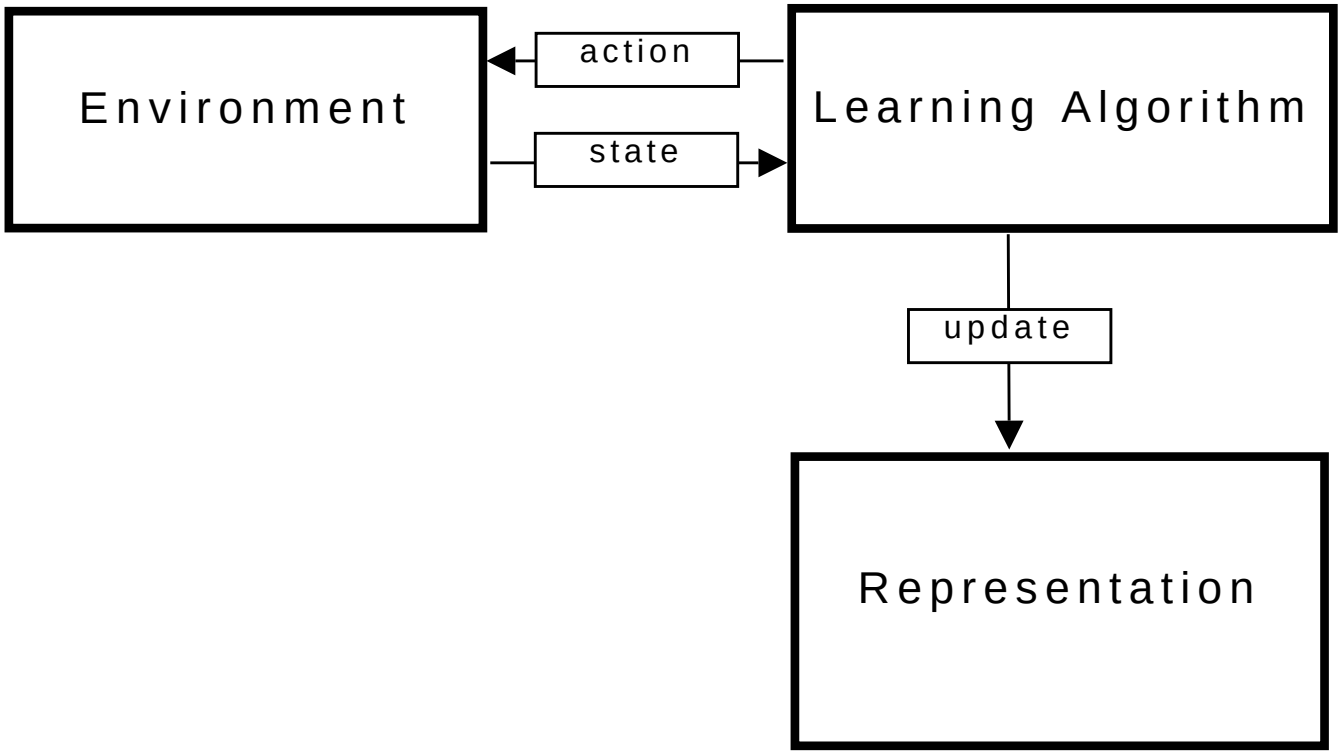
Our Model



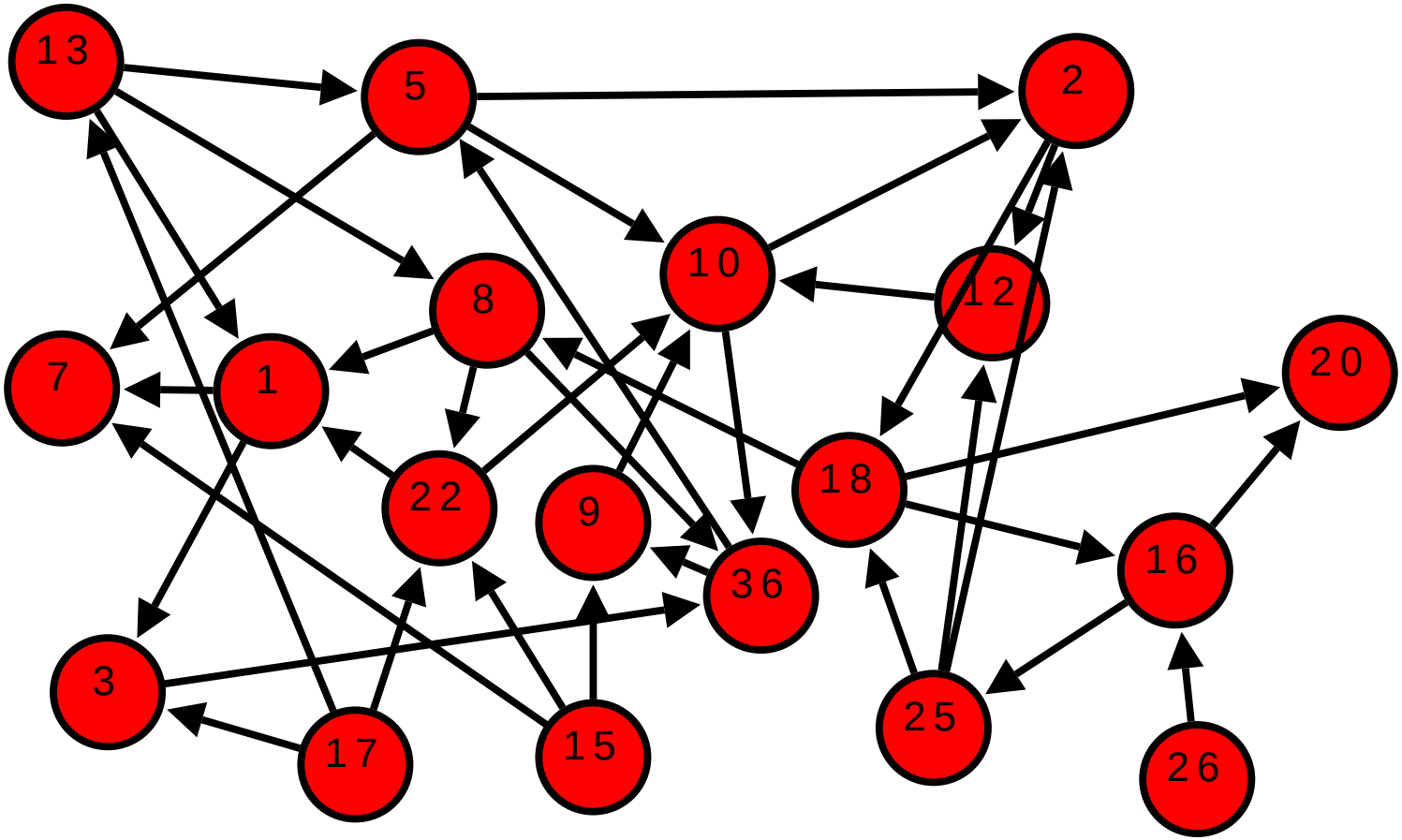
Our Model


Environment
action
state
Learning Algorithm
Representation

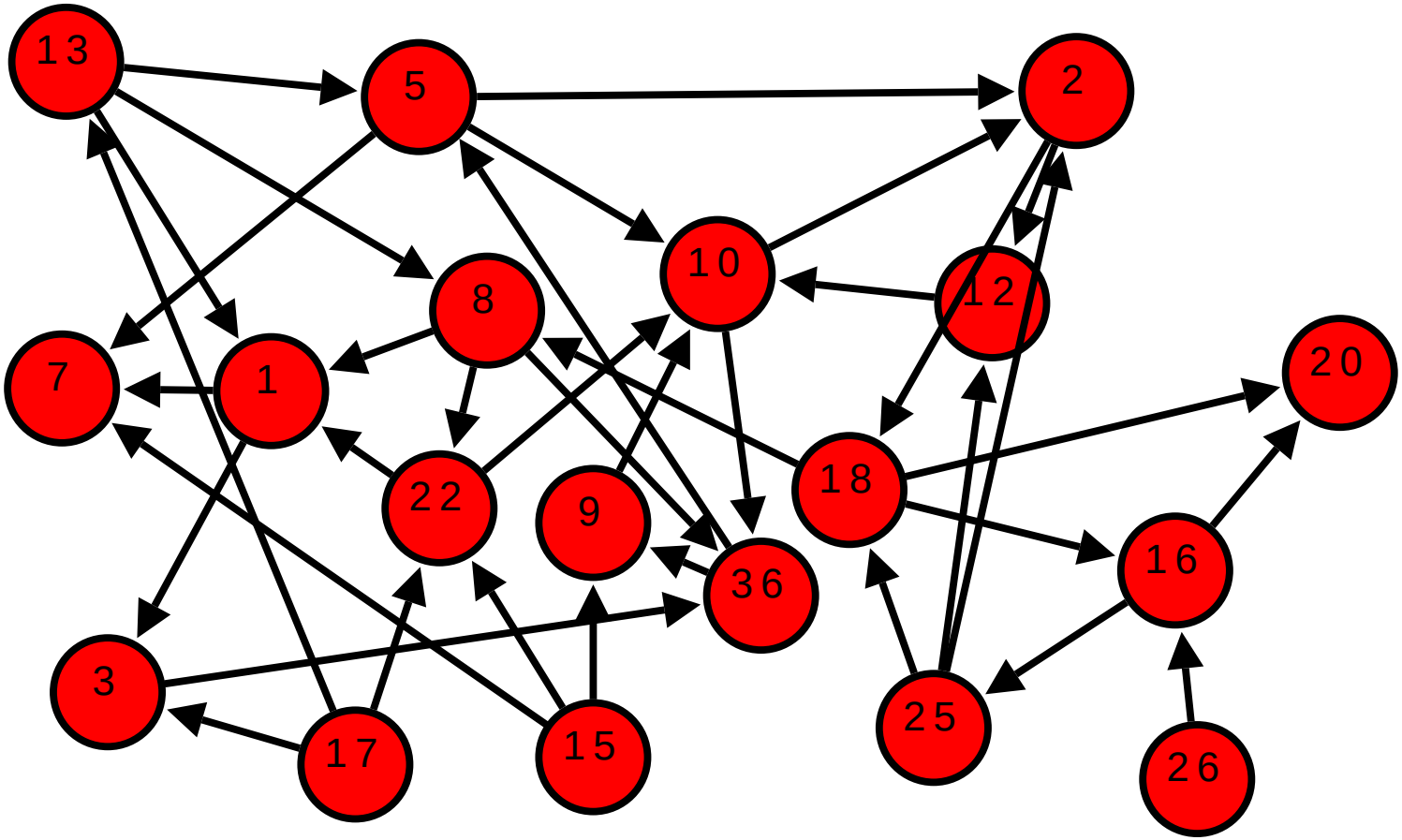
Our Model


Environment
action
state
Learning Algorithm
update
Representation

An environment

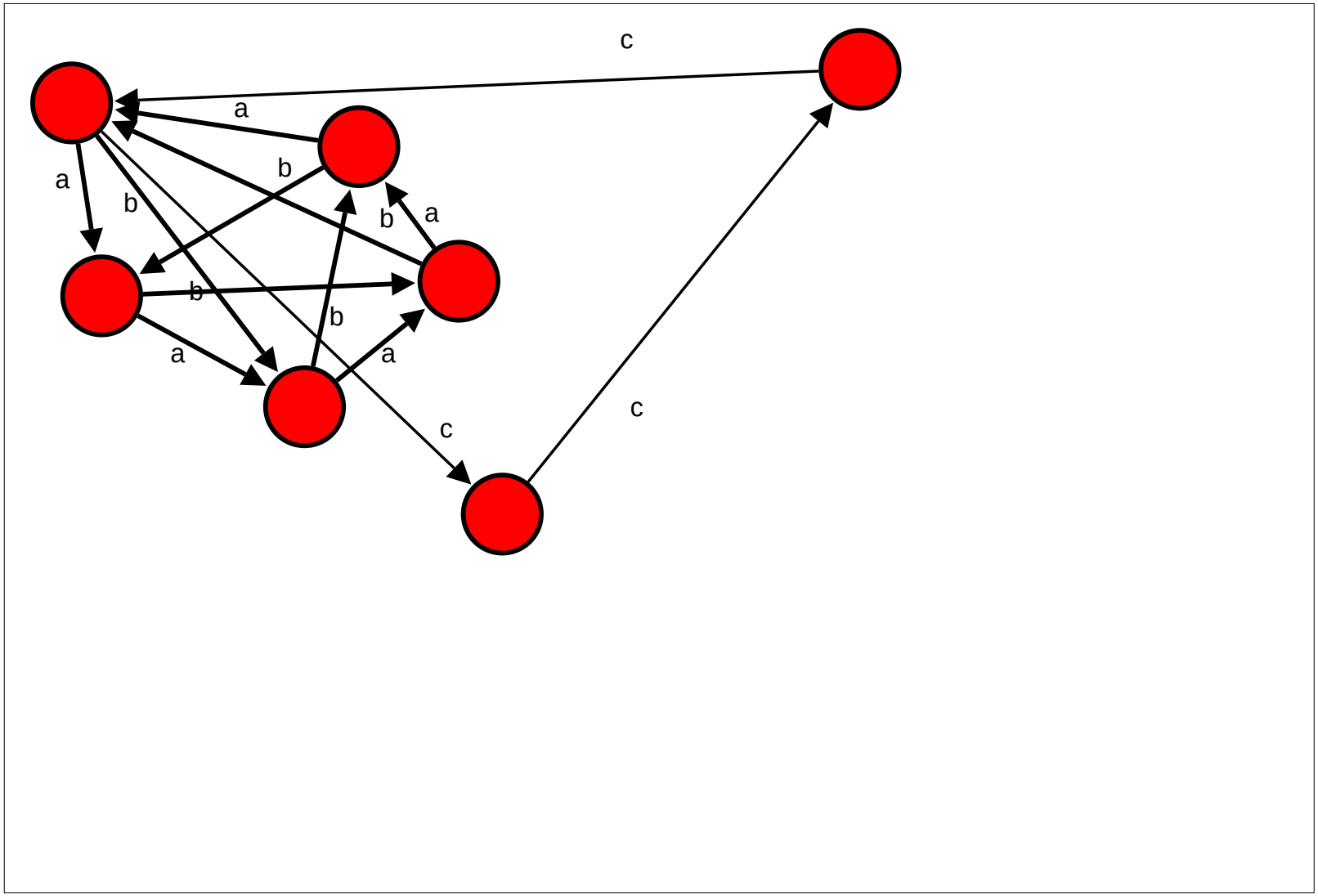


An environment

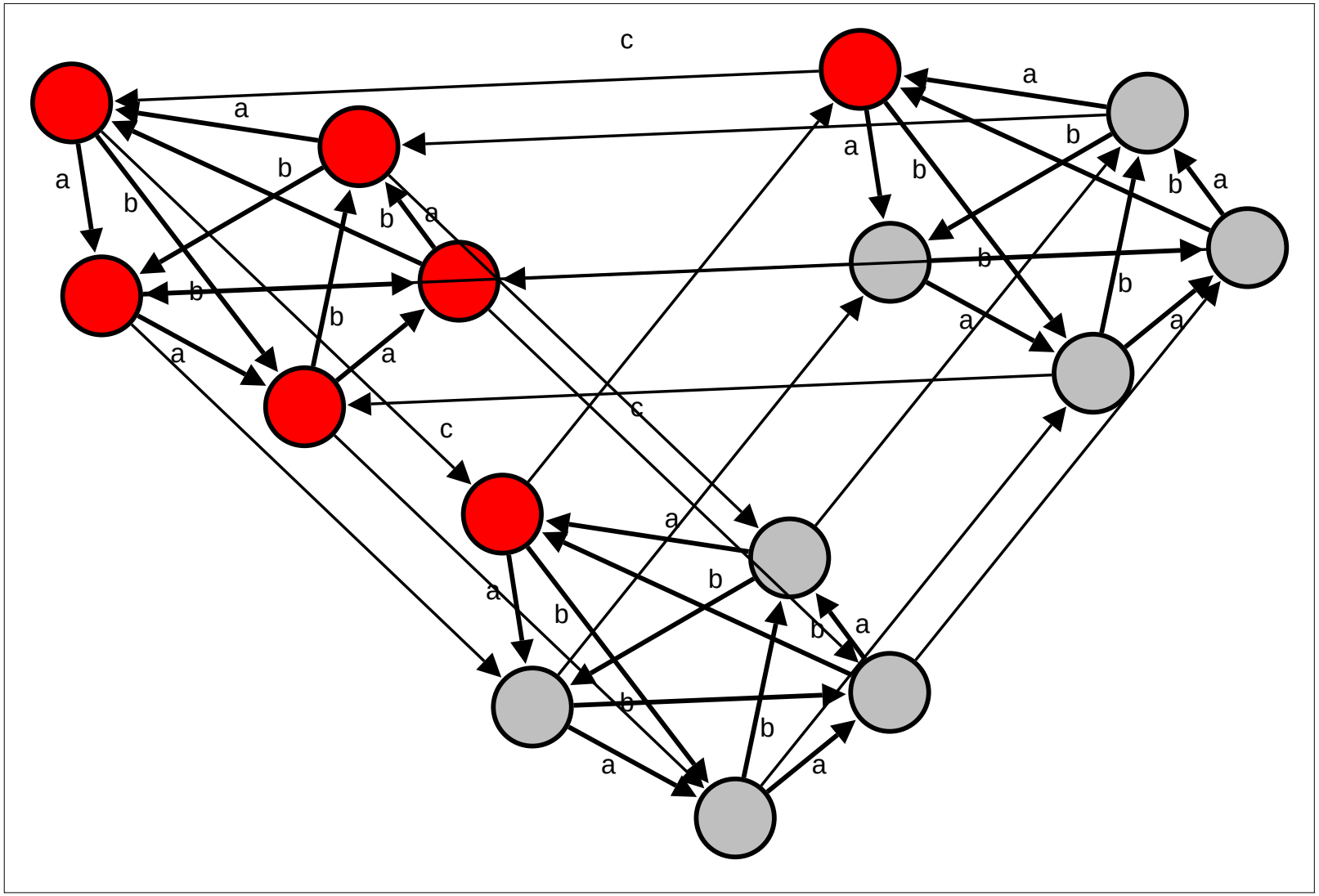


Can we learn the environment without visiting every state?

Example



Example



The structure

- Motivation: Rubik's cube

The structure

- Motivation: Rubik's cube
- It has a *group* structure

The structure

- Motivation: Rubik's cube
- It has a *group* structure
- Groups capture symmetry properties

The structure

- Motivation: Rubik's cube
- It has a *group* structure
- Groups capture symmetry properties
- Most importantly: *homogenity* of actions

The structure

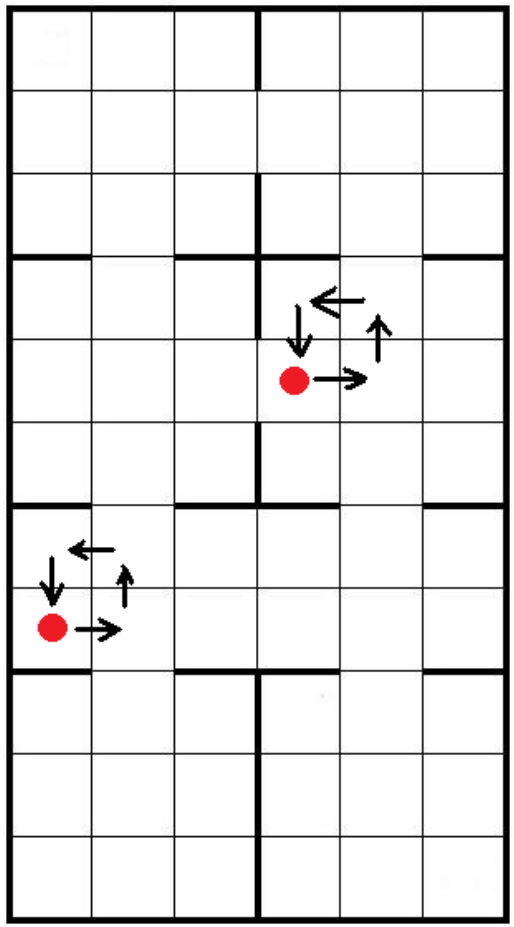
- Motivation: Rubik's cube
- It has a *group* structure
- Groups capture symmetry properties
- Most importantly: *homogenity* of actions

  Initial state is irrelevant

The structure

- Motivation: Rubik's cube
- It has a *group* structure
- Groups capture symmetry properties
- Most importantly: *homogenity* of actions

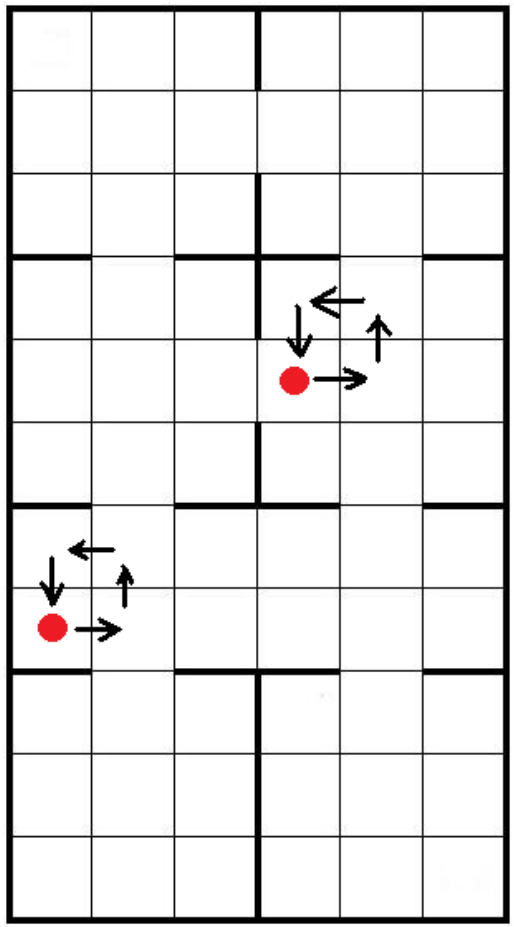
  Initial state is irrelevant
  On the infinite gridworld:
  $\rightarrow\uparrow\leftarrow\downarrow$ leads me back



The structure

- Motivation: Rubik's cube
- It has a *group* structure
- Groups capture symmetry properties
- Most importantly: *homogenity* of actions

  Initial state is irrelevant
  On the infinite gridworld:
  $\rightarrow\uparrow\leftarrow\downarrow$ leads me back
- Makes *generalization* possible



Efficient Learning

Efficient Learning

A class $\mathcal{C}$ of groups is efficiently learnable with respect to a set P of presentations, if:

- After N iterations the learning algorithm stops and returns a decision procedure which is consistent with the target group $\mathcal{G}$
- N is polynomial in $\sum_{a \in A} order(a) + \log |\mathcal{G}|$

Efficient Learning

Efficient Learning

A class $\mathcal{C}$ of groups is efficiently learnable with respect to a set P of presentations, if:

- After N iterations the learning algorithm stops and returns a decision procedure which is consistent with the target group $\mathcal{G}$
- N is polynomial in $\sum_{a \in A} order(a) + \log |\mathcal{G}|$

Note

The set of presentations decides which generators are given as input

Efficient Learning

Efficient Learning

A class $\mathcal{C}$ of groups is efficiently learnable with respect to a set P of presentations, if:

- After N iterations the learning algorithm stops and returns a decision procedure which is consistent with the target group $\mathcal{G}$
- N is polynomial in $\sum_{a \in A} order(a) + \log |\mathcal{G}|$

Note

The set of presentations decides which generators are given as input

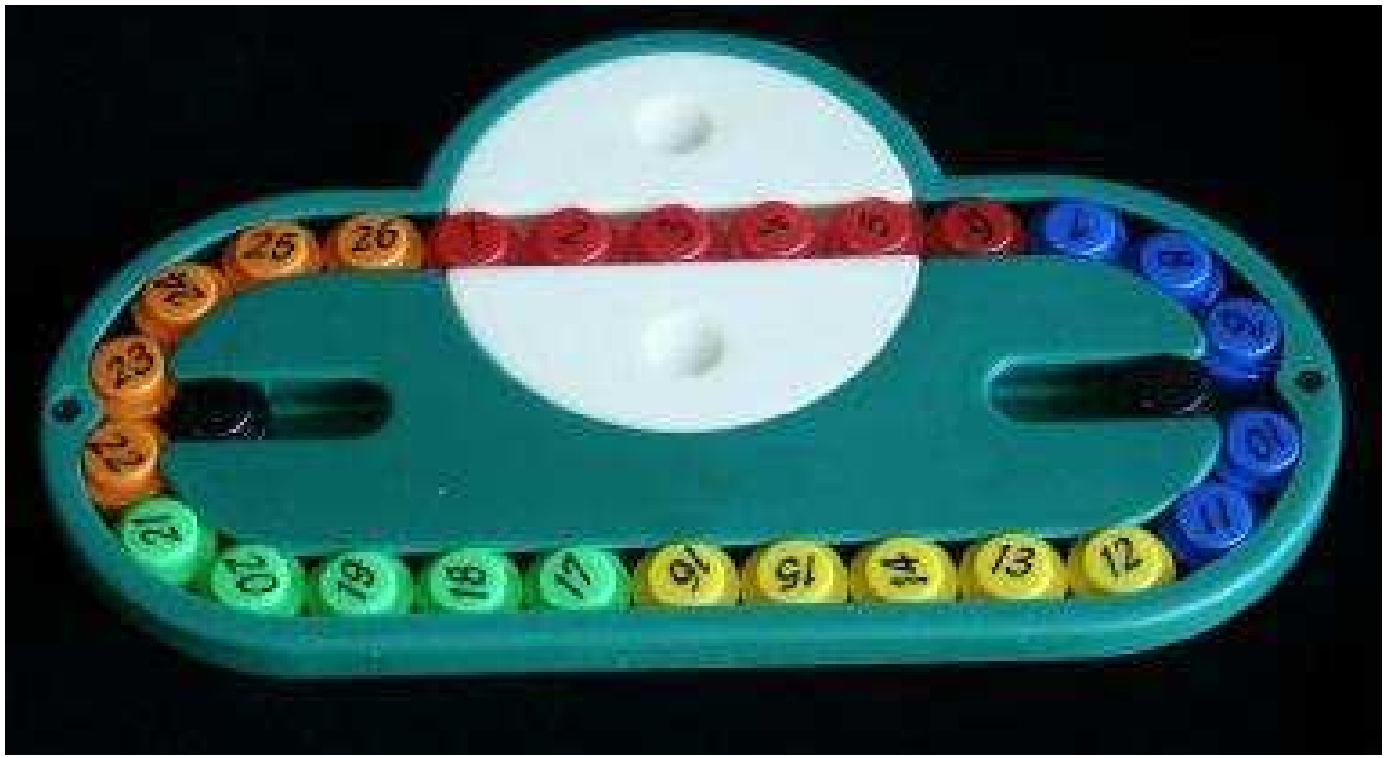
Theorem

The class of all finite groups is **not** *learnable efficiently with respect to the set of all possible presentations.*

Efficiently learnable permutation games

Efficiently learnable permutation games

- Topspin

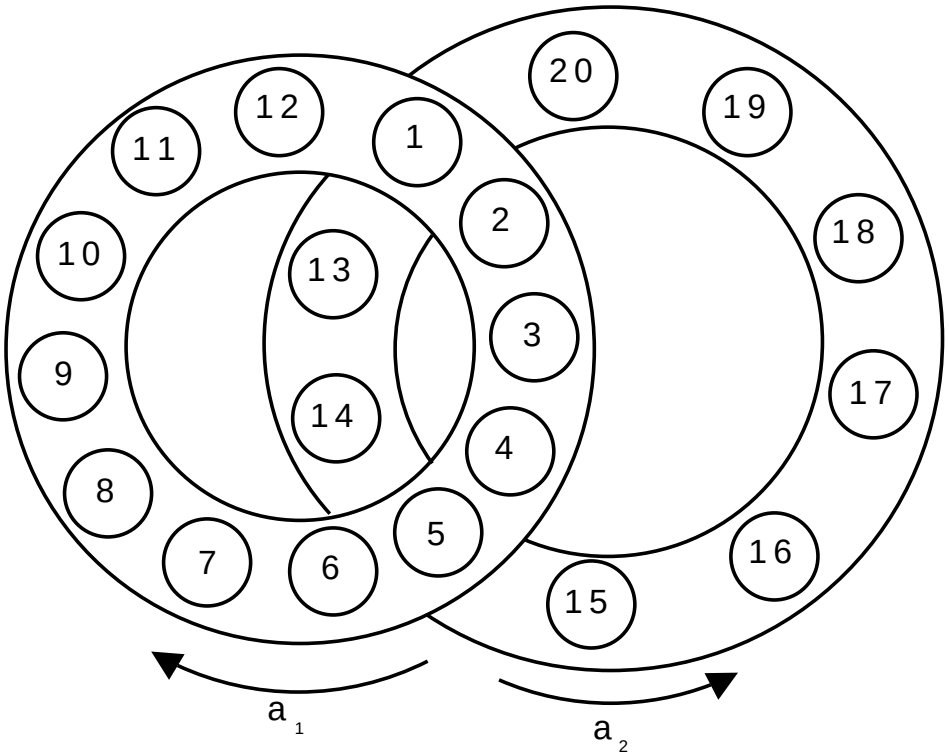


Efficiently learnable permutation games

- Hungarian rings

Efficiently learnable permutation games

- Hungarian rings



An open problem

- Rubik's cube

An open problem

- Rubik's cube

Conclusions

What can we do if the initial representation is not good?

Conclusions

What can we do if the initial representation is not good?

We can try to learn one

Conclusions

What can we do if the initial representation is not good?

We can try to learn one

In general, learning a representation is hard

Conclusions

What can we do if the initial representation is not good?

We can try to learn one

In general, learning a representation is hard

Sometimes we can do it

Conclusions

What can we do if the initial representation is not good?

We can try to learn one

In general, learning a representation is hard

Sometimes we can do it

We can gain a lot by having the right representation

And now...

And now...

Discussion!