Mobile Agent Computing

Example:
Label-Guided Graph Exploration with constant memory

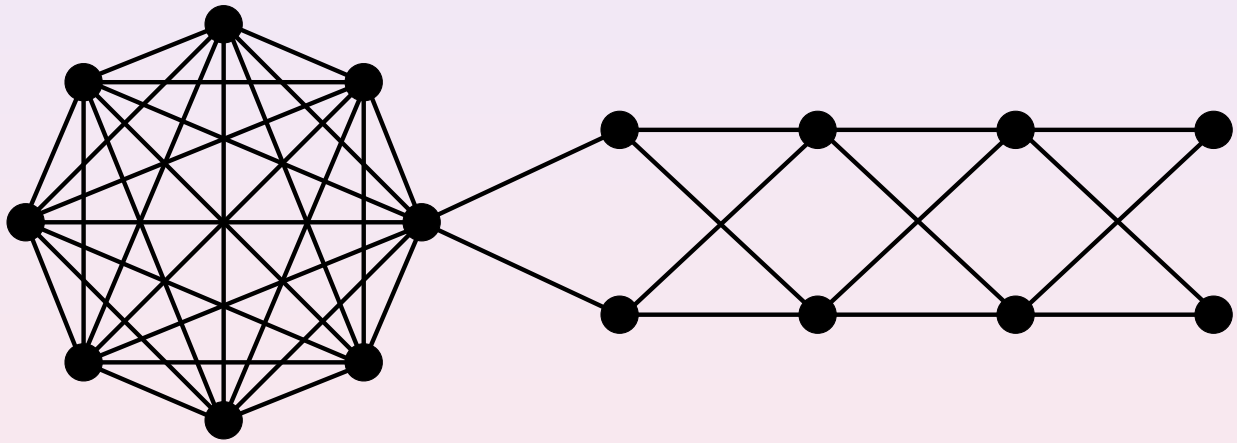
David Ilcinkas

Joint work with
R. Cohen, P. Fraigniaud, A. Korman, and D. Peleg

Graph exploration

Goal

A mobile entity has to traverse every edge of an unknown anonymous graph.



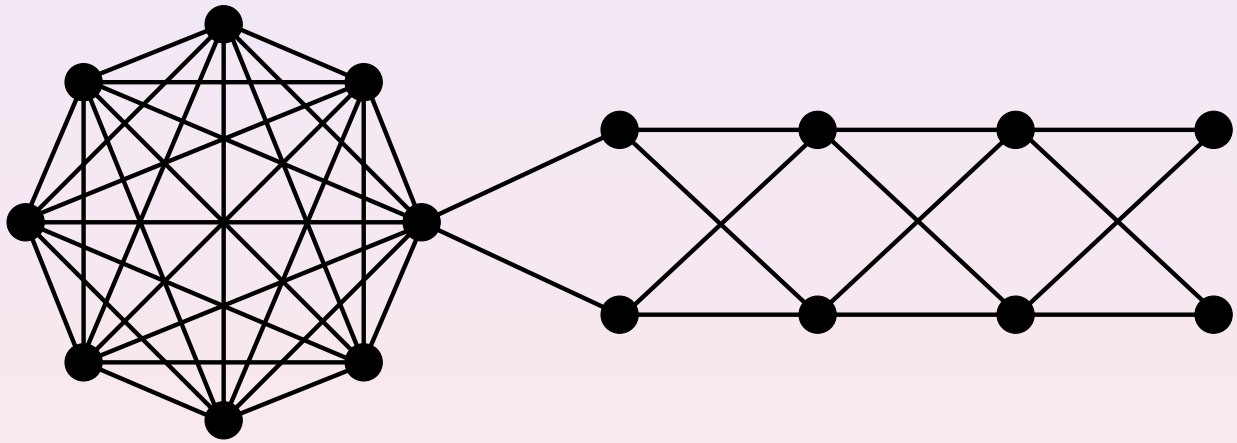
Constraint

The mobile entity should have little memory.

Graph exploration

Goal

A mobile entity has to traverse every edge of an unknown anonymous graph.



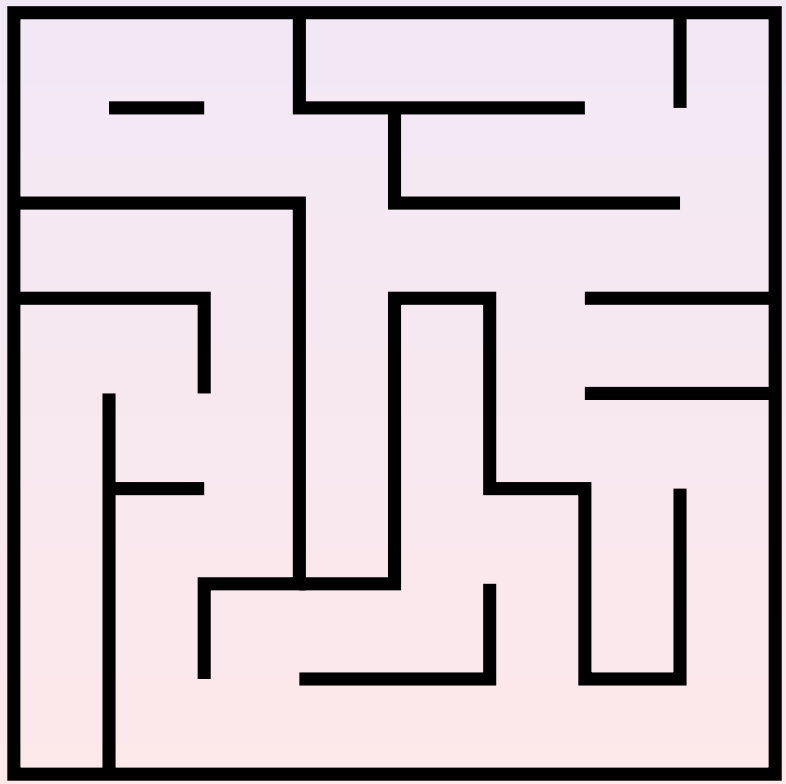
Constraint

The mobile entity should have little memory.

Particular case: Labyrinths

Definition

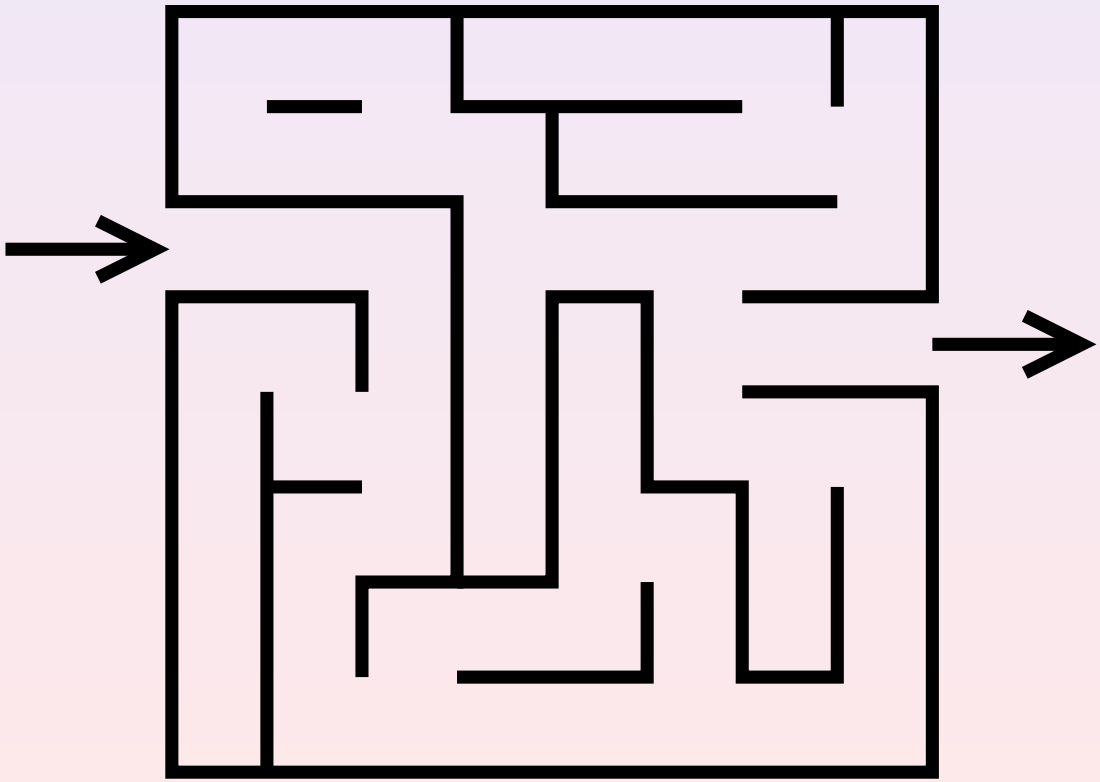
- Grid with missing edges
- Knowledge of North, South, East, West



Particular case: Labyrinths

Definition

- Grid with missing edges
- Knowledge of North, South, East, West



Motivations

Exploration by mobile agents

- Physical robot: exploration of environments unreachable by humans
- Software agent: network maintenance, ressource discovery

Equivalence between logic and automata

- Exploring finite automata with nested pebbles
- First-order logic with transitive closure

Space complexity theory

Graph exploration is related to $L \subseteq SL \subseteq NL$

- Exploration of undirected graphs: complete for SL
- Exploration of directed graphs: complete for NL

Motivations

Exploration by mobile agents

- Physical robot: exploration of environments unreachable by humans
- Software agent: network maintenance, ressource discovery

Equivalence between logic and automata

- Exploring finite automata with nested pebbles
- First-order logic with transitive closure

Space complexity theory

Graph exploration is related to $L \subseteq SL \subseteq NL$

- Exploration of undirected graphs: complete for SL
- Exploration of directed graphs: complete for NL

Motivations

Exploration by mobile agents

- Physical robot: exploration of environments unreachable by humans
- Software agent: network maintenance, ressource discovery

Equivalence between logic and automata

- Exploring finite automata with nested pebbles
- First-order logic with transitive closure

Space complexity theory

Graph exploration is related to $L \subseteq SL \subseteq NL$

- Exploration of undirected graphs: complete for SL
- Exploration of directed graphs: complete for NL

Unknown, anonymous

Unknown

- Unknown topology
- Unknown size (no upper bound)

Anonymous

- No node labeling
- Local edge labeling

Unknown, anonymous

Unknown

- Unknown topology
- Unknown size (no upper bound)

Anonymous

- No node labeling
- Local edge labeling

Unknown, anonymous

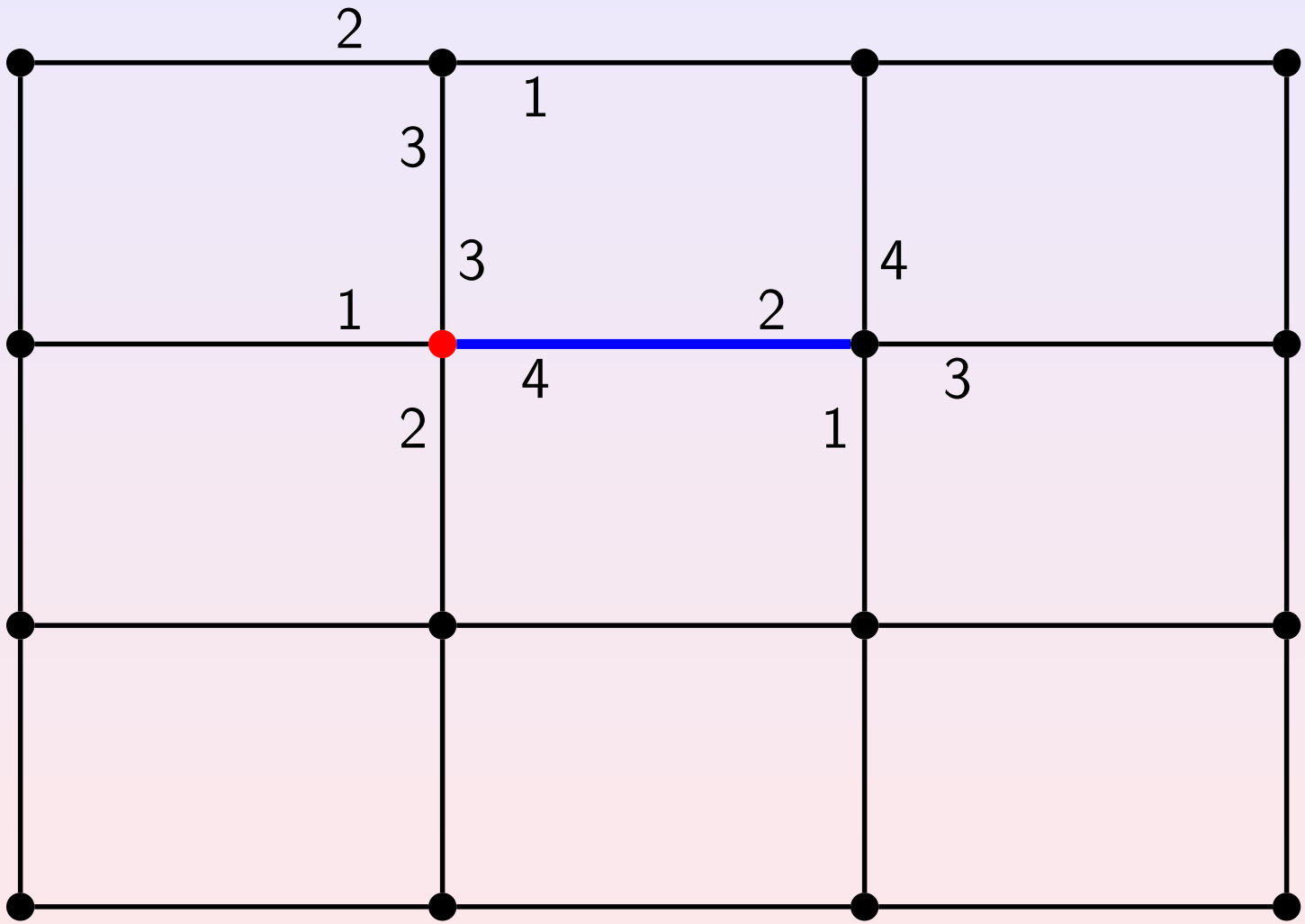
Unknown

- Unknown topology
- Unknown size (no upper bound)

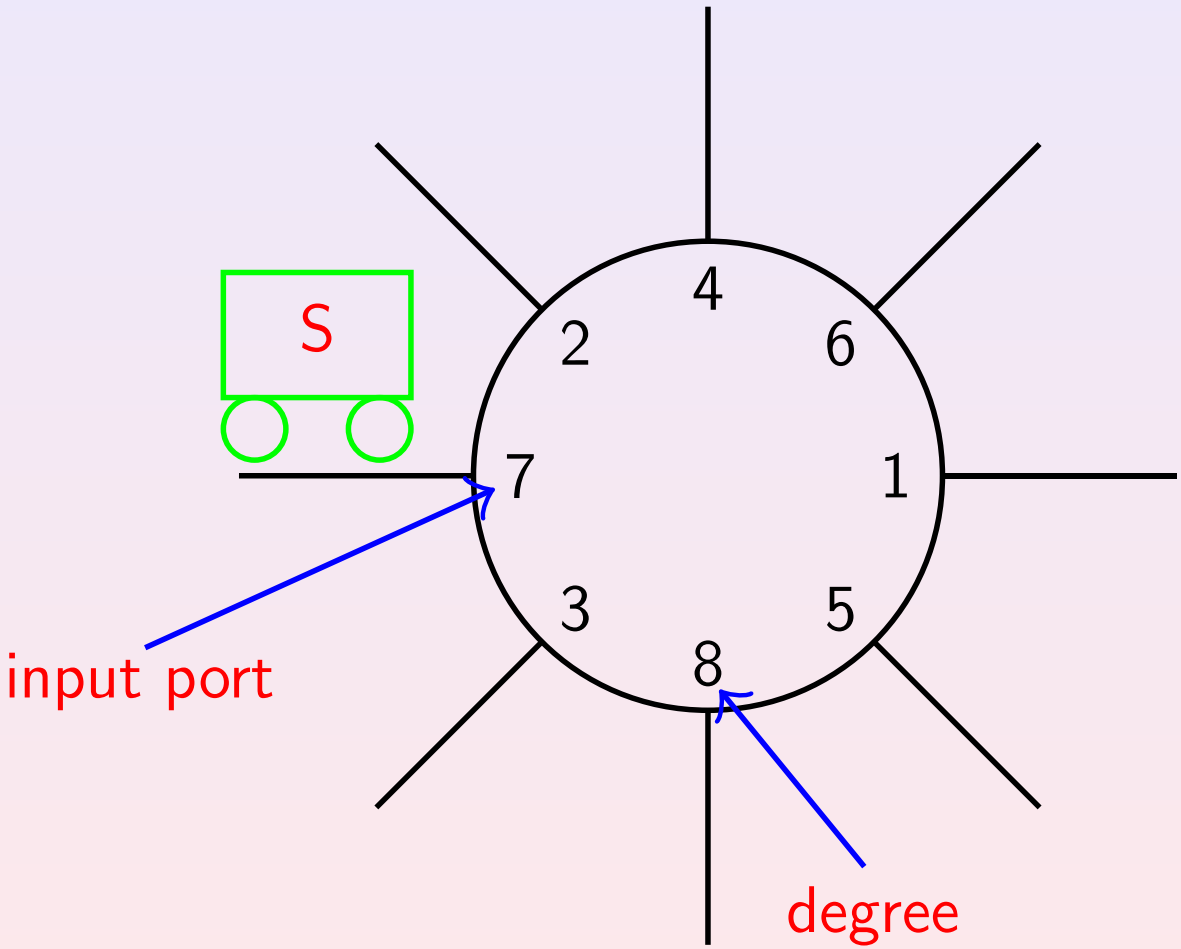
Anonymous

- No node labeling
- Local edge labeling

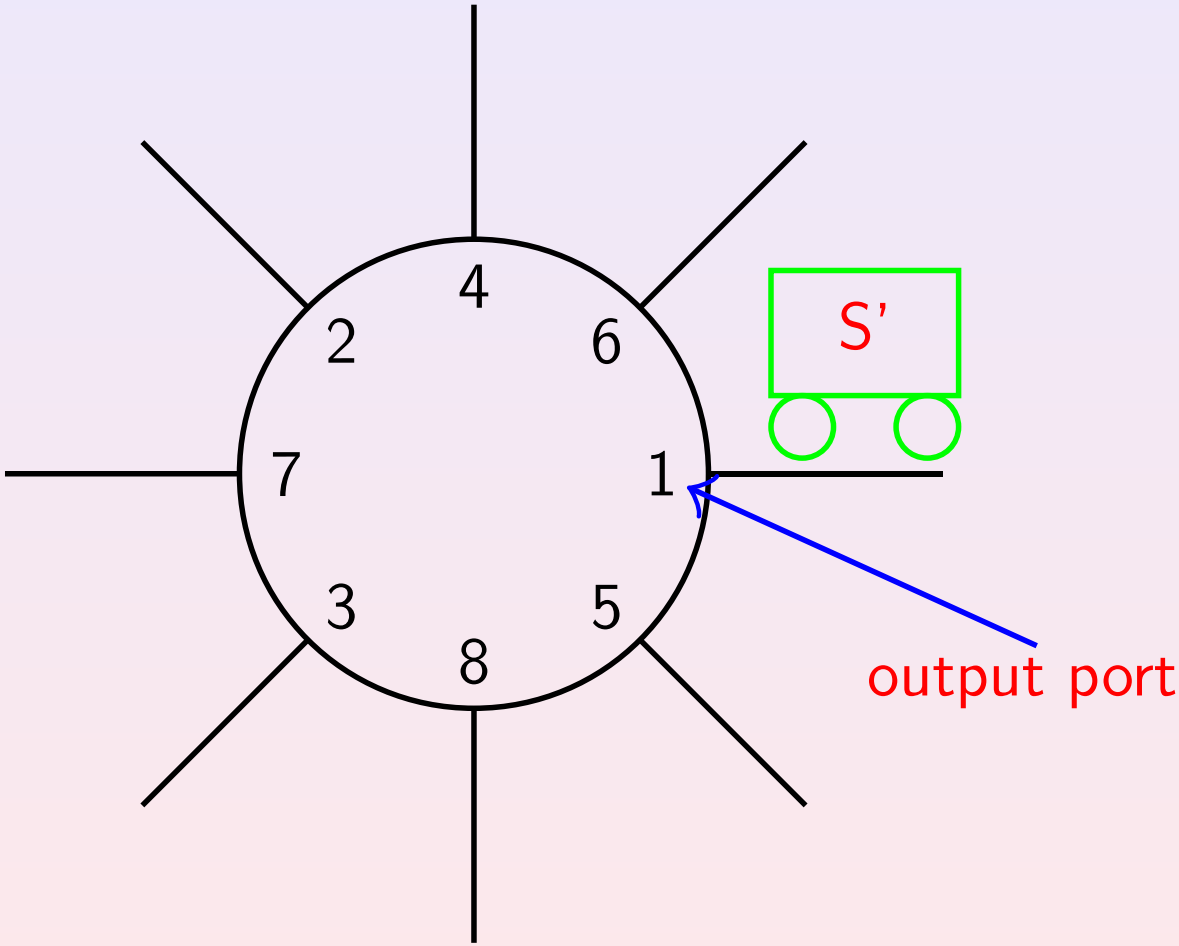
Example of an anonymous graph



Mealy automaton (1)



Mealy automaton (1)



Mealy automaton (2)

Input

- S : current state
- i : input port number
- d : node's degree

Transition function

$(S', j) = \mu\ (S, i, d)$

Output

- S' : new state
- j : output port number

Impossibility results

Budach, Math. Nachrichten, 1978
Automata and Labyrinths

No finite automaton can explore all graphs.

A JAG (Jumping Automaton for Graphs) is a team of finite automata that cooperate constantly. Moreover an automaton can jump to a vertex occupied by another automaton.

Cook, Rackoff, SIAMJC, 1980
Space lower bounds for maze threadability on restricted machines

No JAG can explore all graphs.

Impossibility results

Budach, Math. Nachrichten, 1978
Automata and Labyrinths

No finite automaton can explore all graphs.

A JAG (Jumping Automaton for Graphs) is a team of finite automata that cooperate constantly. Moreover an automaton can jump to a vertex occupied by another automaton.

Cook, Rackoff, SIAMJC, 1980
Space lower bounds for maze threadability on restricted machines

No JAG can explore all graphs.

Impossibility results

Budach, Math. Nachrichten, 1978
Automata and Labyrinths

No finite automaton can explore all graphs.

A JAG (Jumping Automaton for Graphs) is a team of finite automata that cooperate constantly. Moreover an automaton can jump to a vertex occupied by another automaton.

Cook, Rackoff, SIAMJC, 1980
Space lower bounds for maze threadability on restricted machines

No JAG can explore all graphs.

Our model

Model

- An oracle colors (labels) the graph to help the automaton.
- The finite automaton can read the color of the node as an input of its transition function.

Goal

Use the smallest possible number of colors.

Our results

Theorem 1: Three colors

There exist a finite automaton and an algorithm coloring in three colors such that the automaton can explore all graphs.

Theorem 2: Two colors

There exist an automaton of $O(log\Delta)$ memory bits and an algorithm coloring in only two colors such that the automaton can explore all graphs of maximum degree Δ.

Our results

Theorem 1: Three colors

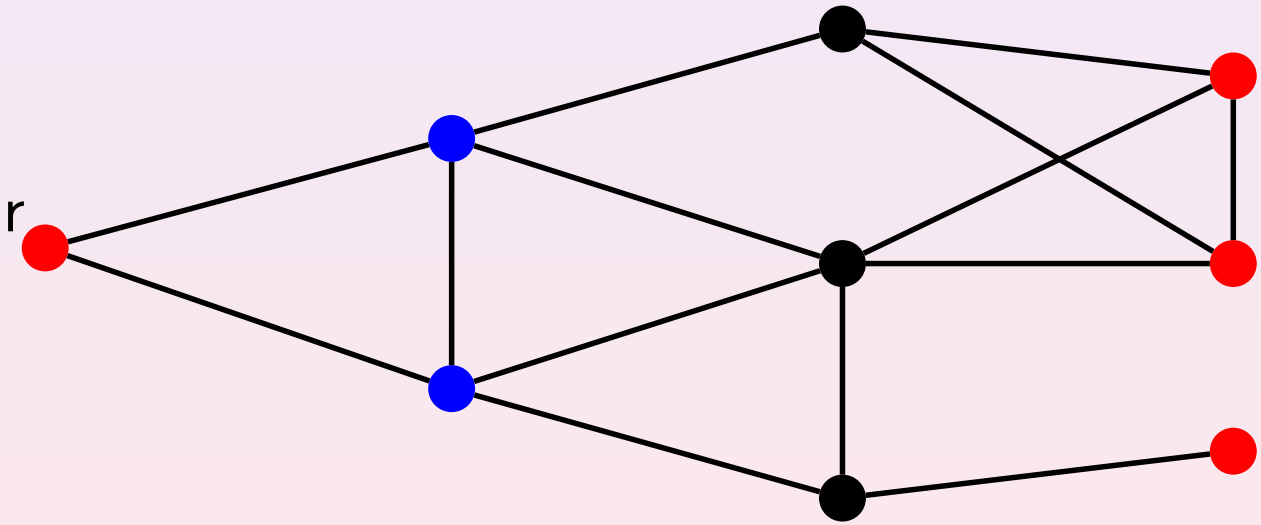
There exist a finite automaton and an algorithm coloring in three colors such that the automaton can explore all graphs.

Theorem 2: Two colors

There exist an automaton of $O(log\,\Delta)$ memory bits and an algorithm coloring in only two colors such that the automaton can explore all graphs of maximum degree Δ.

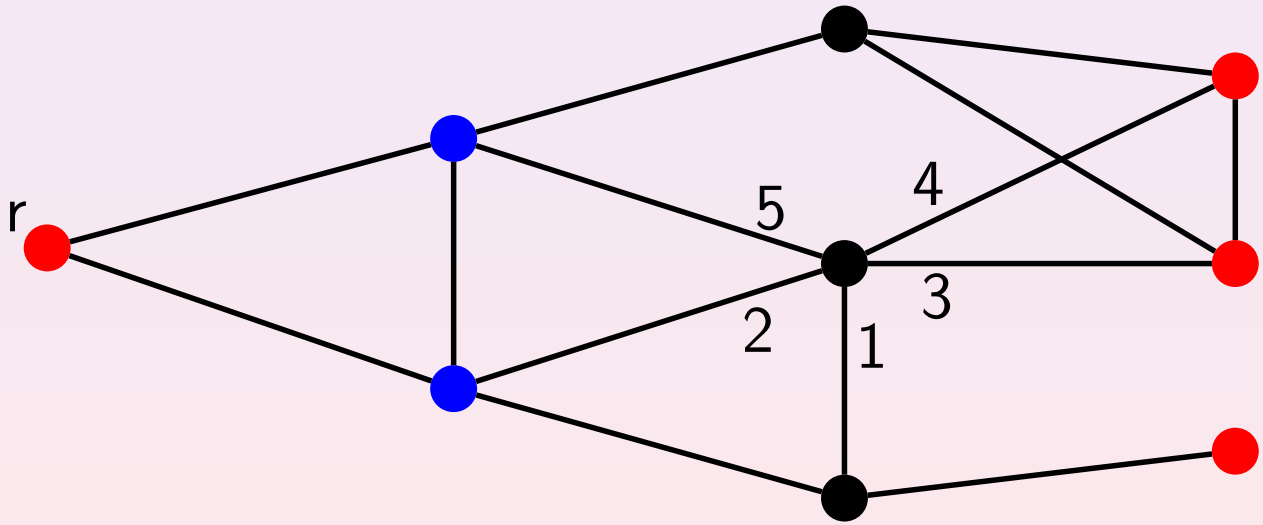
Three colors are enough

- choose arbitrarily a node as the root
- color all nodes according to their distance d to the root
  - distance $d \cong 0[n]$ red
  - distance $d \cong 1[n]$ blue
  - distance $d \cong 2[n]$ black



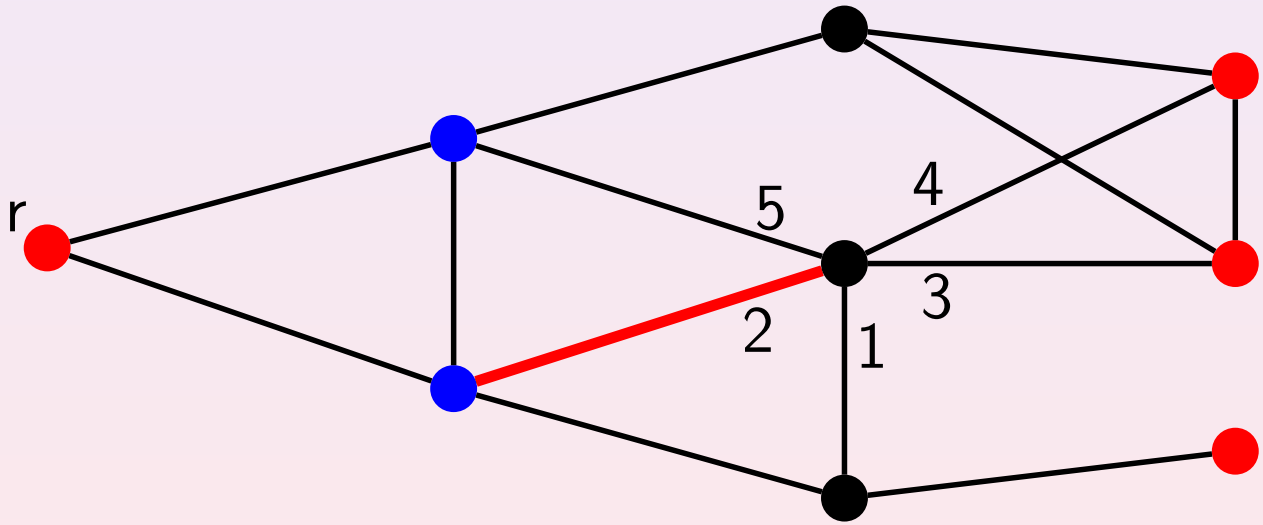
Three colors are enough

- choose arbitrarily a node as the root
- color all nodes according to their distance d to the root
  - distance $d \cong 0[n]$ red
  - distance $d \cong 1[n]$ blue
  - distance $d \cong 2[n]$ black



Three colors are enough

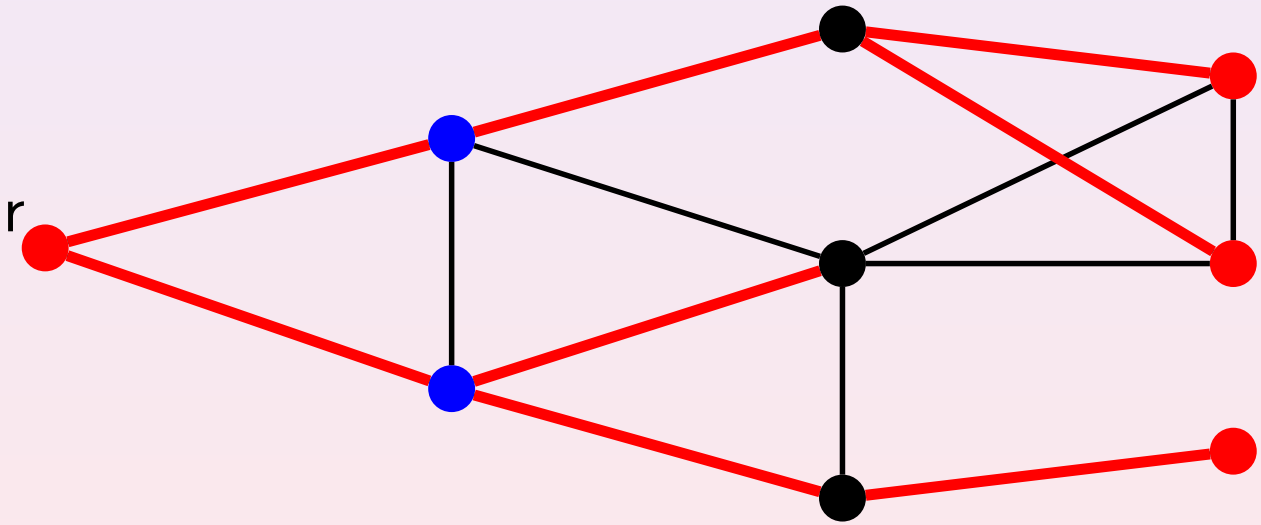
- choose arbitrarily a node as the root
- color all nodes according to their distance d to the root
  - distance $d \cong 0[n]$ red
  - distance $d \cong 1[n]$ blue
  - distance $d \cong 2[n]$ black



The smallest port number leading to the above layer defines the parent in the spanning tree.

Three colors are enough

- choose arbitrarily a node as the root
- color all nodes according to their distance d to the root
  - distance $d \cong 0[n]$ red
  - distance $d \cong 1[n]$ blue
  - distance $d \cong 2[n]$ black



The smallest port number leading to the above layer defines the parent in the spanning tree.

Only two colors?

- Layer 1: red
- Layer 2: blue
- Layer 3: red
- Layer 4: red
- Layer 5: red
- Layer 6: blue
- Layer 7: blue
- Layer 8: blue

Open problem

Conclusion

- Three colors are sufficient for arbitrary graphs.
- Two colors are necessary and sufficient for graphs of constant degree.

Open problem

Are two colors sufficient for arbitrary graphs?

Open problem

Conclusion

- Three colors are sufficient for arbitrary graphs.
- Two colors are necessary and sufficient for graphs of constant degree.

Open problem

Are two colors sufficient for arbitrary graphs?