

Space Lower Bounds for Graph Exploration via Reduced Automata

Pierre Fraigniaud¹ David Ilcinkas² Sergio Rajsbaum³
Sébastien Tixeuil²

¹CNRS, LRI, Université Paris-Sud, France

²LRI, Université Paris-Sud, France

³Instituto de Matemáticas, UNAM, D. F. 04510, Mexico

SIROCCO '05

May 24, 2005

Graph exploration

Goal

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

Motivation

- exploration of environments unreachable by humans
- network maintenance
- map drawing

Graph exploration

Goal

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

Motivation

- exploration of environments unreachable by humans
- network maintenance
- map drawing

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

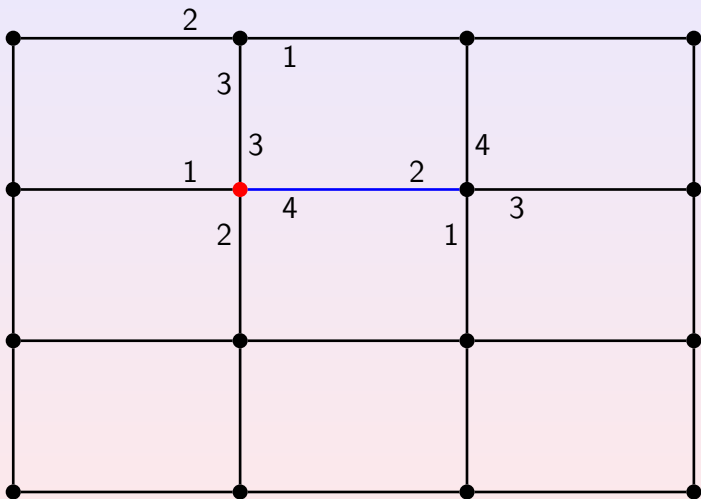
Unknown

- Unknown topology
- Unknown size (no upper bound)

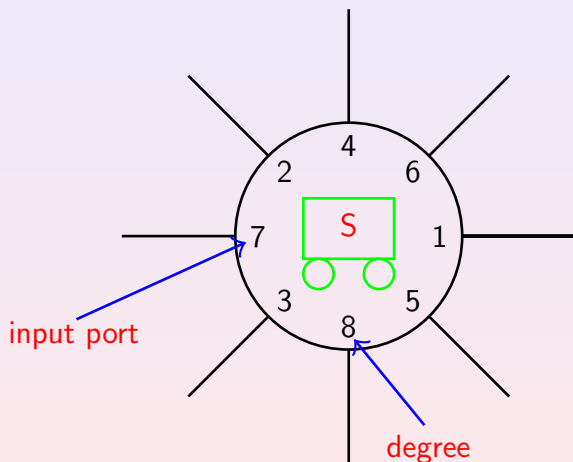
Anonymous

- No node labeling
- Local edge labeling

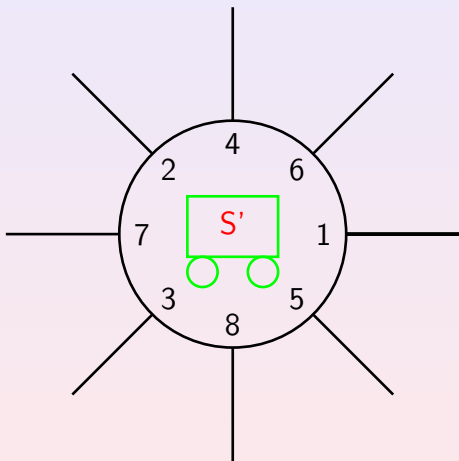
Example of an anonymous graph



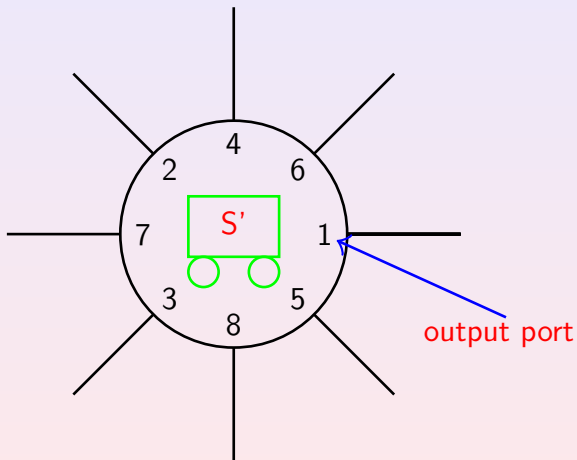
Moore automaton



Moore automaton



Moore automaton



Plan

- 1 Introduction
- 2 Related work
 - Perpetual exploration
 - Exploration with stop
 - Our results
- 3 Sketch of the proofs
- 4 Conclusion

Exploration by a single automaton

Budach, Math. Nachrichten, 1978

Automata and Labyrinths

No finite automaton can explore all graphs.

MFC5 2004

Graph exploration by a finite automaton

Exploration by a single automaton

Budach, Math. Nachrichten, 1978

Automata and Labyrinths

No finite automaton can explore all graphs.

Fraigniaud, Ilcinkas, Peer, Pelc, Peleg, MFCS 2004

Graph exploration by a finite automaton

- For any K -state automaton, there exists a trap of at most $K + 1$ nodes.
- An automaton which can explore all graphs of at most n nodes has at least n states ($\Omega(\log n)$ memory bits).
- DFS is space optimal: $\Theta(D \log \Delta)$ bits in the family of graphs with diameter at most D and degree at most Δ .

Exploration by a single automaton

Budach, Math. Nachrichten, 1978

Automata and Labyrinths

No finite automaton can explore all graphs.

Fraigniaud, Ilcinkas, Peer, Pelc, Peleg, MFCS 2004

Graph exploration by a finite automaton

- For any K -state automaton, there exists a trap of at most $K + 1$ nodes.
- An automaton which can explore all graphs of at most n nodes has at least n states ($\Omega(\log n)$ memory bits).
- DFS is space optimal: $\Theta(D \log \Delta)$ bits in the family of graphs with diameter at most D and degree at most Δ

Exploration by a single automaton

Budach, Math. Nachrichten, 1978

Automata and Labyrinths

No finite automaton can explore all graphs.

Fraigniaud, Ilcinkas, Peer, Pelc, Peleg, MFCS 2004

Graph exploration by a finite automaton

- For any K -state automaton, there exists a trap of at most $K + 1$ nodes.
- An automaton which can explore all graphs of at most n nodes has at least n states ($\Omega(\log n)$ memory bits).
- **DFS is space optimal**: $\Theta(D \log \Delta)$ bits in the family of graphs with diameter at most D and degree at most Δ

Collective exploration

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

No finite team of finite cooperative automata can explore all graphs.

Our result: Theorem 1

For any set of q non-cooperative K -state automata there exists a graph of size $O(qK)$ that these automata cannot explore.

(Rollik: size $O(K^{O(q)})$ for the same set)

Collective exploration

Rollik, Acta Informatica, 1980
Automaten in planaren Graphen

No finite team of finite cooperative automata can explore all graphs.

Our result: Theorem 1

For any set of q non-cooperative K -state automata there exists a graph of size $O(qK)$ that these automata cannot explore.

(Rollik: size $O(K^{O(q)})$ for the same set)

Exploration with stop

The robot is required **to stop** after completing exploration.

Observation

A single robot cannot explore the ring and stop.

The robot can drop a **pebble** in a node and later identify it and pick it up.

Chen, IEEE TRA 1991

Robotic exploration as graph construction:

- Map drawing with one pebble in time $O(mn)$

Exploration with stop

The robot is required **to stop** after completing exploration.

Observation

A single robot cannot explore the ring and stop.

The robot can drop a pebble in a node and later identify it and pick it up.

IEEE TRA 1991

Robotic exploration as graph construction

- Map drawing with one pebble in time $O(mn)$

Exploration with stop

The robot is required **to stop** after completing exploration.

Observation

A single robot cannot explore the ring and stop.

The robot can drop a **pebble** in a node and later identify it and pick it up.

IEEE TRA 1991

Robotic exploration as graph construction

- Map drawing with one pebble in time $O(mn)$

Exploration with stop

The robot is required **to stop** after completing exploration.

Observation

A single robot cannot explore the ring and stop.

The robot can drop a **pebble** in a node and later identify it and pick it up.

Dudek, Jenkin, Milios, Wilkes, IEEE TRA 1991

Robotic exploration as graph construction

- Map drawing with one pebble in time $O(mn)$

Our results

Theorem 2 (implied by Theorem 1)

For any K -state automaton with one pebble there exists a trap of size $O(K^3)$.

Corollary

An automaton which performs exploration with stop in all graphs of at most n nodes has at least $\Omega(\log n)$ bits of memory.

Theorem 3

There exists an exploration with stop algorithm which requires $O(D \log \Delta)$ bits of memory when performed in the family of graphs with diameter at most D and degree at most Δ .

Our results

Theorem 2 (implied by Theorem 1)

For any K -state automaton with one pebble there exists a trap of size $O(K^3)$.

Corollary

An automaton which performs exploration with stop in all graphs of at most n nodes has at least $\Omega(\log n)$ bits of memory.

Theorem 3

There exists an exploration with stop algorithm which requires $O(D \log \Delta)$ bits of memory when performed in the family of graphs with diameter at most D and degree at most Δ .

Our results

Theorem 2 (implied by Theorem 1)

For any K -state automaton with one pebble there exists a trap of size $O(K^3)$.

Corollary

An automaton which performs exploration with stop in all graphs of at most n nodes has at least $\Omega(\log n)$ bits of memory.

Theorem 3

There exists an exploration with stop algorithm which requires $O(D \log \Delta)$ bits of memory when performed in the family of graphs with diameter at most D and degree at most Δ .

Perpetual vs. With stop

Both models: $\Omega(\log n)$ lower bound

- Perpetual: without pebble
- With stop: with pebble

Alfred A. Chou, SODA 2002

Tree exploration with little memory

- Perpetual: $\Theta(\log \Delta)$ bits
- With stop: $\Omega(\log \log \log n)$ bits

Perpetual vs. With stop

Both models: $\Omega(\log n)$ lower bound

- Perpetual: without pebble
- With stop: with pebble

Diks, Fraigniaud, Kranakis, Pelc, SODA 2002

Tree exploration with little memory

- Perpetual: $\Theta(\log \Delta)$ bits
- With stop: $\Omega(\log \log \log n)$ bits

Plan

- 1 Introduction
- 2 Related work
- 3 Sketch of the proofs**
 - Reduced automaton
 - Non-cooperative: $O(qK)$
 - With stop: $\Omega(\log n)$
 - With stop: $O(D \log \Delta)$
- 4 Conclusion

Homogeneous graphs

d -homogeneous graph

- d -regular
- edge-colored (same label at both extremities)

In d -homogeneous graphs:

- $S' = f(S, i, d)$ (Moore automaton)
- $S' = f(S, g(S), d)$ (edge-colored)
- $S' = h(S)$ (d -regular)

Homogeneous graphs

d -homogeneous graph

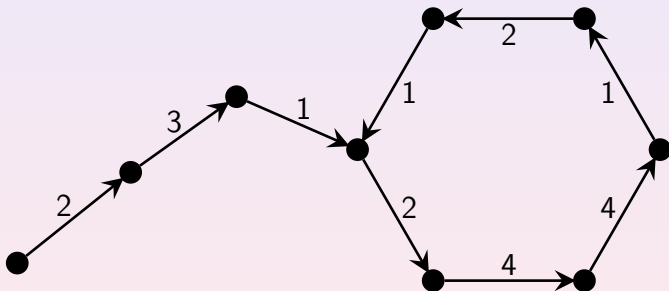
- d -regular
- edge-colored (same label at both extremities)

In d -homogeneous graphs:

- $S' = f(S, i, d)$ (Moore automaton)
- $S' = f(S, g(S), d)$ (edge-colored)
- $S' = h(S)$ (d -regular)

Reduced automaton (1)

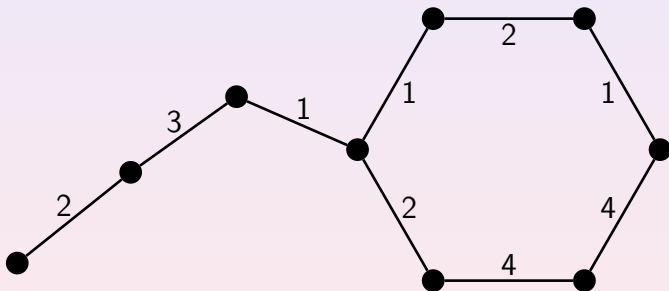
In d -homogeneous graphs the transition graph of the automaton is a 1-factor.



Footprint of the automaton: $231(244121)^\infty$

Reduced automaton (1)

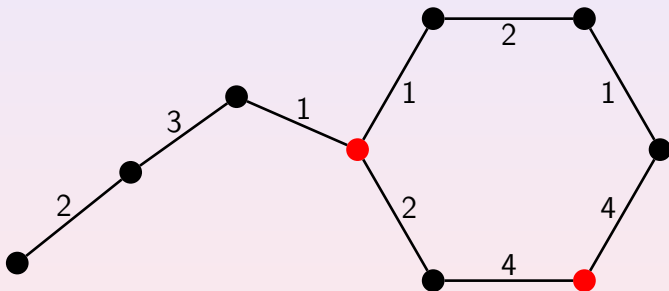
In d -homogeneous graphs the transition graph of the automaton is a 1-factor.



Footprint of the automaton: $231(244121)^\infty$

Reduced automaton (1)

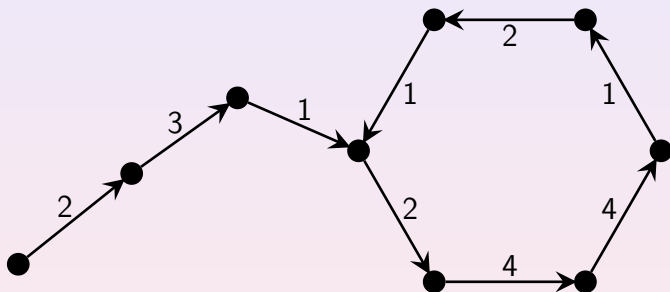
In d -homogeneous graphs the transition graph of the automaton is a 1-factor.



Footprint of the automaton: $231(244121)^\infty$

Reduced automaton (2)

In edge-colored graphs the two moves i, i form a closed walk.

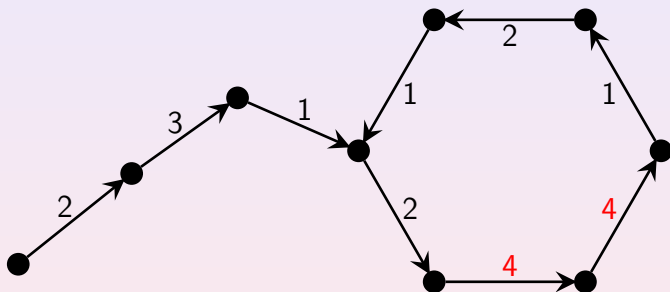


Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $231(244121)^\infty$

Reduced automaton (2)

In edge-colored graphs the two moves i, i form a closed walk.

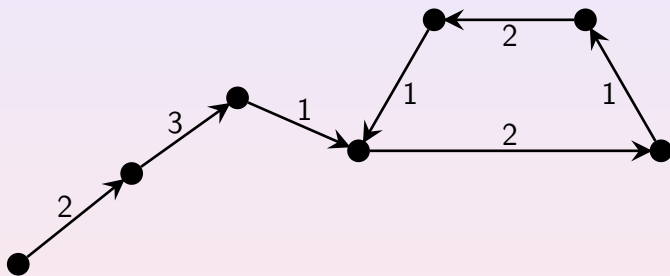


Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $231(244121)^\infty$

Reduced automaton (2)

In edge-colored graphs the two moves i, i form a closed walk.

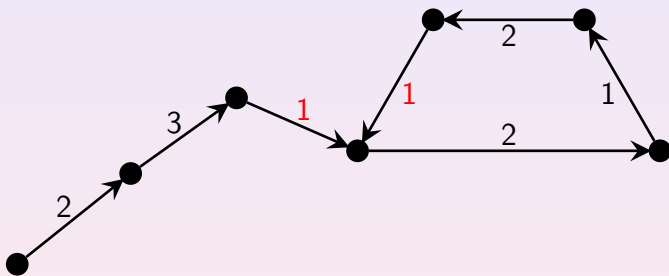


Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $231(2121)^\infty$

Reduced automaton (2)

In edge-colored graphs the two moves i, i form a closed walk.

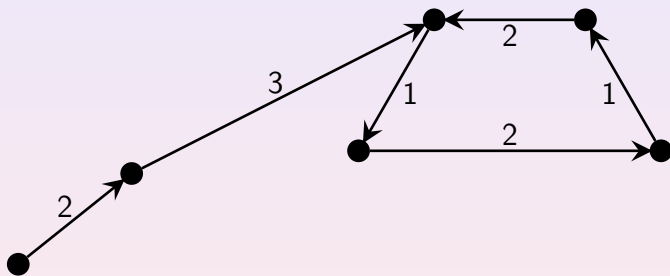


Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $231(2121)^\infty$

Reduced automaton (2)

In edge-colored graphs the two moves i, i form a closed walk.

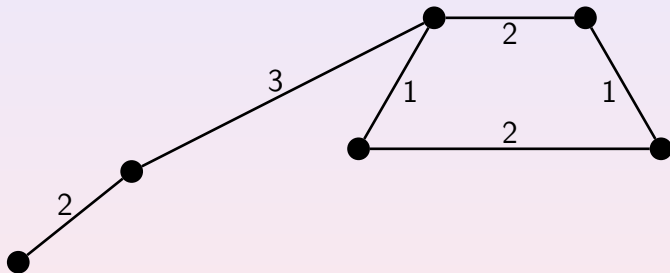


Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $23(1212)^\infty$

Reduced automaton (2)

In edge-colored graphs the two moves i, i form a closed walk.

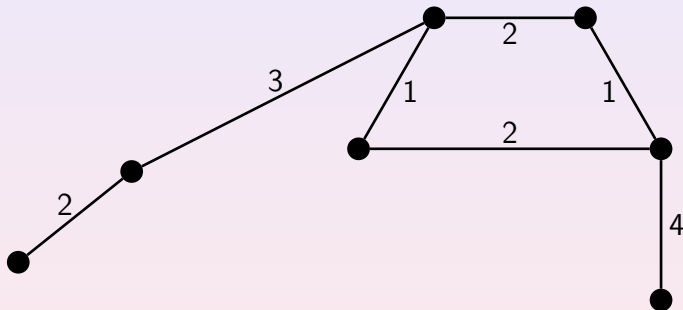


Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $23(1212)^\infty$

Reduced automaton (2)

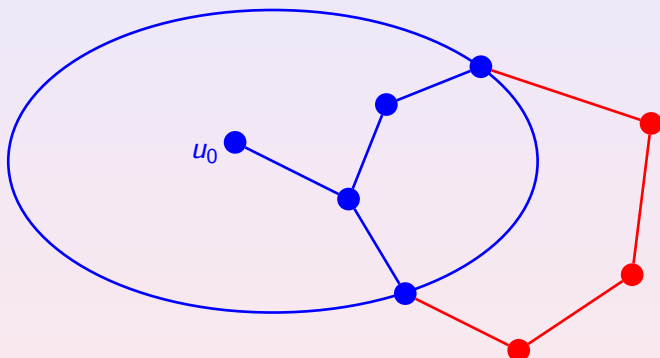
In edge-colored graphs the two moves i, i form a closed walk.



Footprint of the automaton: $231(244121)^\infty$

Footprint of the reduced automaton: $23(1212)^\infty$

Induction



Trap for q non-cooperative K -state automata: $O(qK)$ nodes.

Exploration with stop (1)

Goal

To design a trap for a K -state automaton A with the help of a pebble.

Notations

- A_0 : A without pebble is equivalent to a basic K -state automaton A_0
- A_1 : A always with its pebble is equivalent to a basic K -state automaton A_1

Exploration with stop (2)

- If the automaton is far from the pebble
 - Traps for basic automata (like A_0)
 - The automaton may enter the traps with different states
 - Trap for the K different automata with the same transition function as $A_0 \implies O(K^2)$ nodes
- If the pebble is always close to the automaton
 - Not really more powerful as a basic automaton
 - Trap for A_1 as a meta-structure $\implies O(K)$ nodes

Total size of the trap: $O(K^3)$ nodes

Exploration with stop (2)

- If the automaton is far from the pebble
 - Traps for basic automata (like A_0)
 - The automaton may enter the traps with different states
 - Trap for the K different automata with the same transition function as $A_0 \implies O(K^2)$ nodes
- If the pebble is always close to the automaton
 - Not really more powerful as a basic automaton
 - Trap for A_1 as a meta-structure $\implies O(K)$ nodes

Total size of the trap: $O(K^3)$ nodes

Exploration with stop (2)

- If the automaton is far from the pebble
 - Traps for basic automata (like A_0)
 - The automaton may enter the traps with different states
 - Trap for the K different automata with the same transition function as $A_0 \implies O(K^2)$ nodes
- If the pebble is always close to the automaton
 - Not really more powerful as a basic automaton
 - Trap for A_1 as a meta-structure $\implies O(K)$ nodes

Total size of the trap: $O(K^3)$ nodes

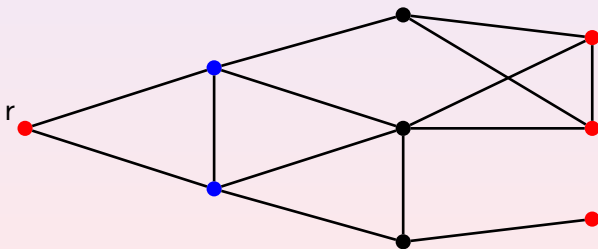
Exploration with stop (2)

- If the automaton is far from the pebble
 - Traps for basic automata (like A_0)
 - The automaton may enter the traps with different states
 - Trap for the K different automata with the same transition function as $A_0 \implies O(K^2)$ nodes
- If the pebble is always close to the automaton
 - Not really more powerful as a basic automaton
 - Trap for A_1 as a meta-structure $\implies O(K)$ nodes

Total size of the trap: $O(K^3)$ nodes

Exploration with stop algorithm

- Based on DFS
- The pebble stays at the root
- A node is at distance at least x from the pebble *iff* the pebble is unreachable by a DFS at depth $x - 1$



Plan

- 1 Introduction
- 2 Related work
- 3 Sketch of the proofs
- 4 Conclusion**

Conclusion

Conclusion

- Trap for a set of q non-cooperative K -state automata: size $O(qK)$ nodes
- $\Omega(\log n)$ lower bound for exploration **with stop**
- $O(D \log \Delta)$ exploration **with stop** algorithm

Open problems

- Optimality of the $\Omega(\log n)$ lower bounds (perpetual and with stop)
- Is $O(D \log \Delta)$ optimal for exploration with stop as it is for perpetual exploration?

Conclusion

Conclusion

- Trap for a set of q non-cooperative K -state automata: size $O(qK)$ nodes
- $\Omega(\log n)$ lower bound for exploration **with stop**
- $O(D \log \Delta)$ exploration **with stop** algorithm

Open problems

- Optimality of the $\Omega(\log n)$ lower bounds (perpetual and with stop)
- Is $O(D \log \Delta)$ optimal for exploration with stop as it is for perpetual exploration?