

Digraphs Exploration with Little Memory

Pierre FRAIGNIAUD
David ILCINKAS (speaker)

CNRS, LRI, Université Paris-Sud, France

Graph Exploration

- *Goal*

a mobile entity in an unknown graph has to

- **traverse** all edges
- **stop** after completing exploration

Graph Exploration

- **Goal**

a mobile entity in an unknown graph has to

- **traverse** all edges
- **stop** after completing exploration

- **Motivations**

- exploration or map drawing of places unreachable by humans
- network maintenance
- searching for data (P2P, Web)

Anonymous graphs

Anonymous graphs

- Unknown size and topology

Anonymous graphs

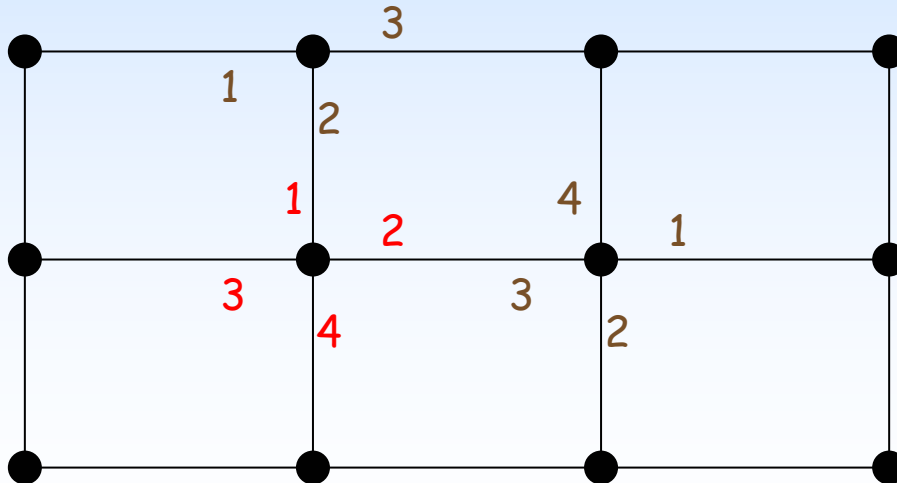
- Unknown size and topology
- No node labeling

Anonymous graphs

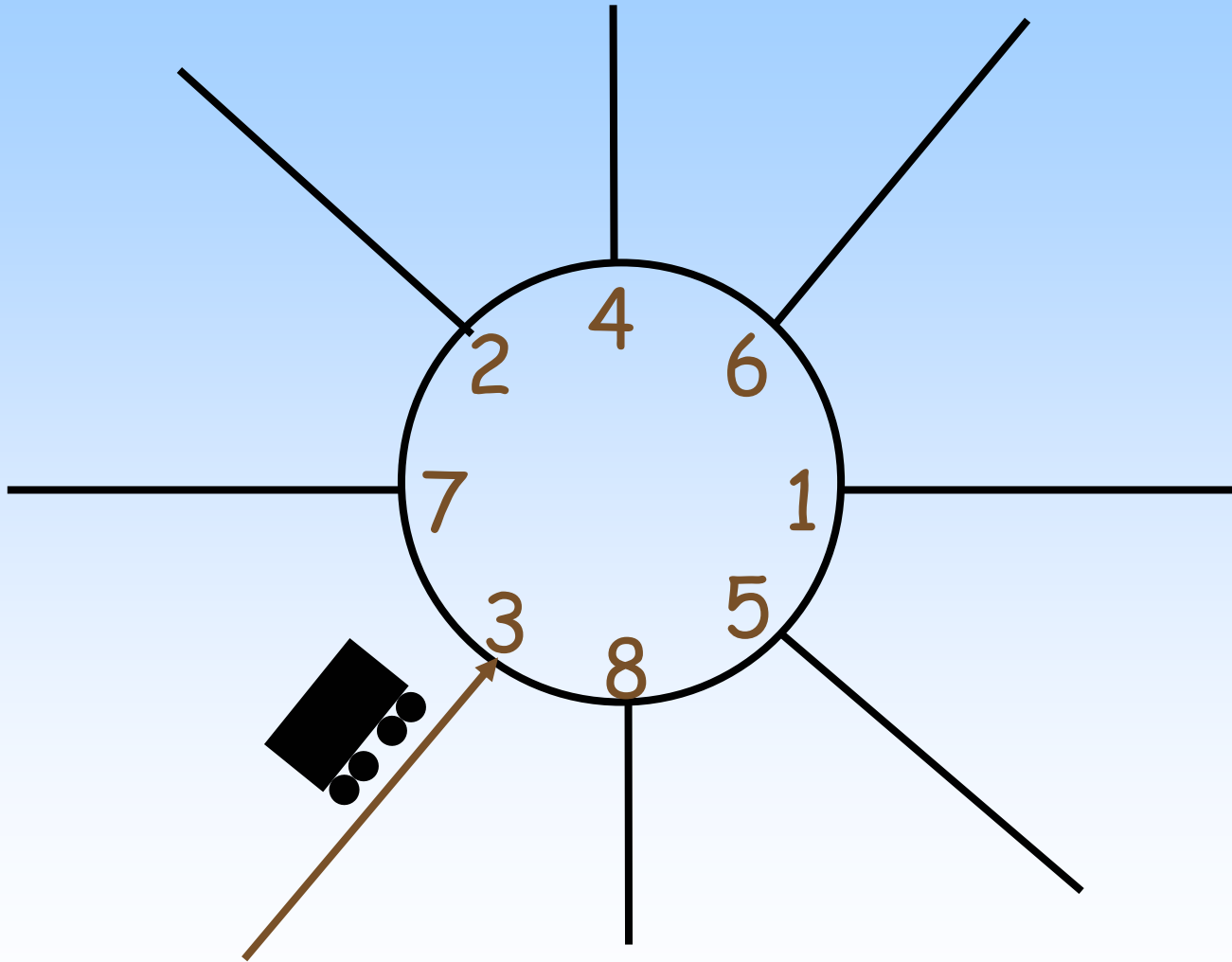
- Unknown size and topology
- No node labeling
- Local edge labeling

Anonymous graphs

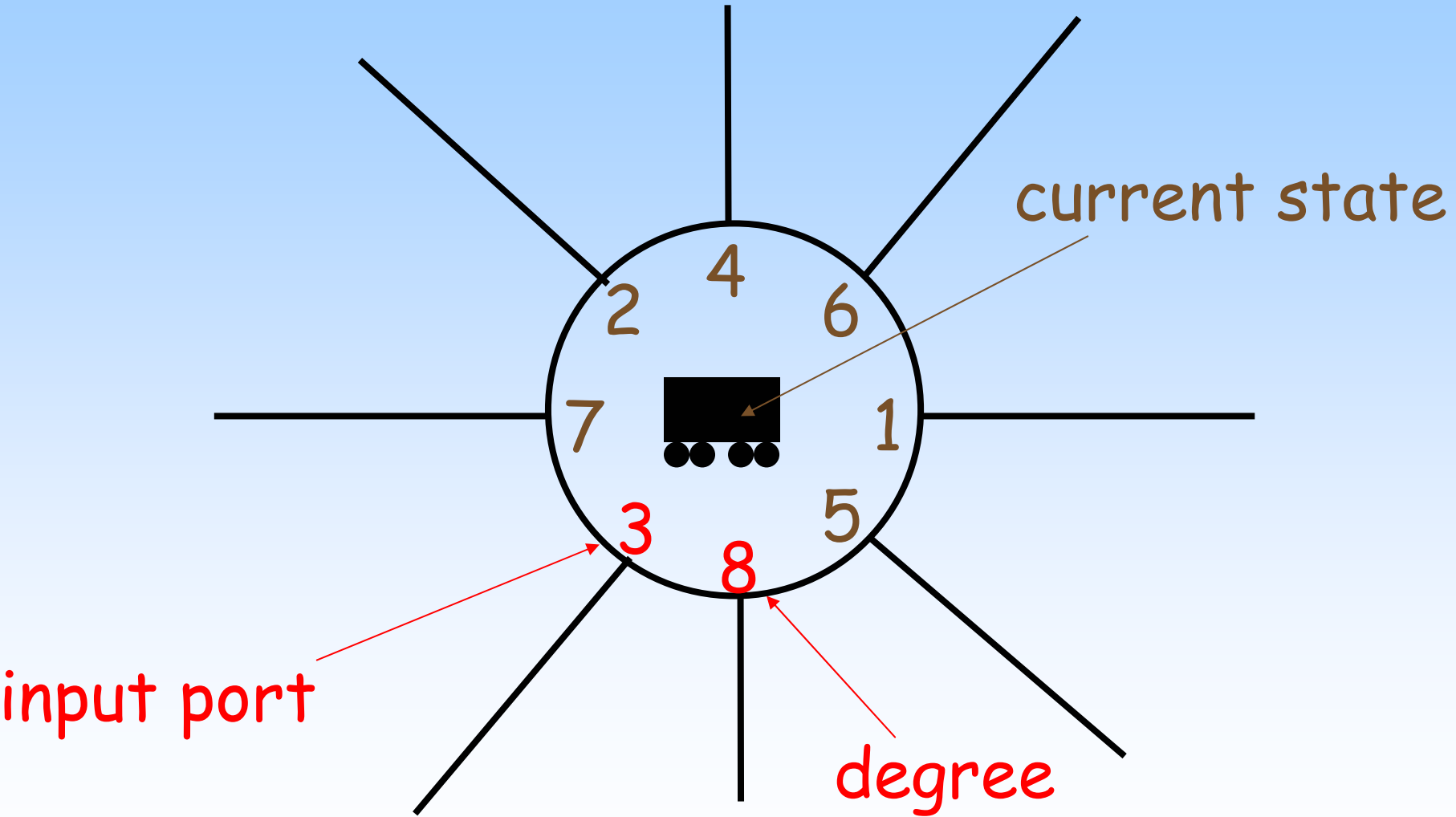
- Unknown size and topology
- No node labeling
- Local edge labeling



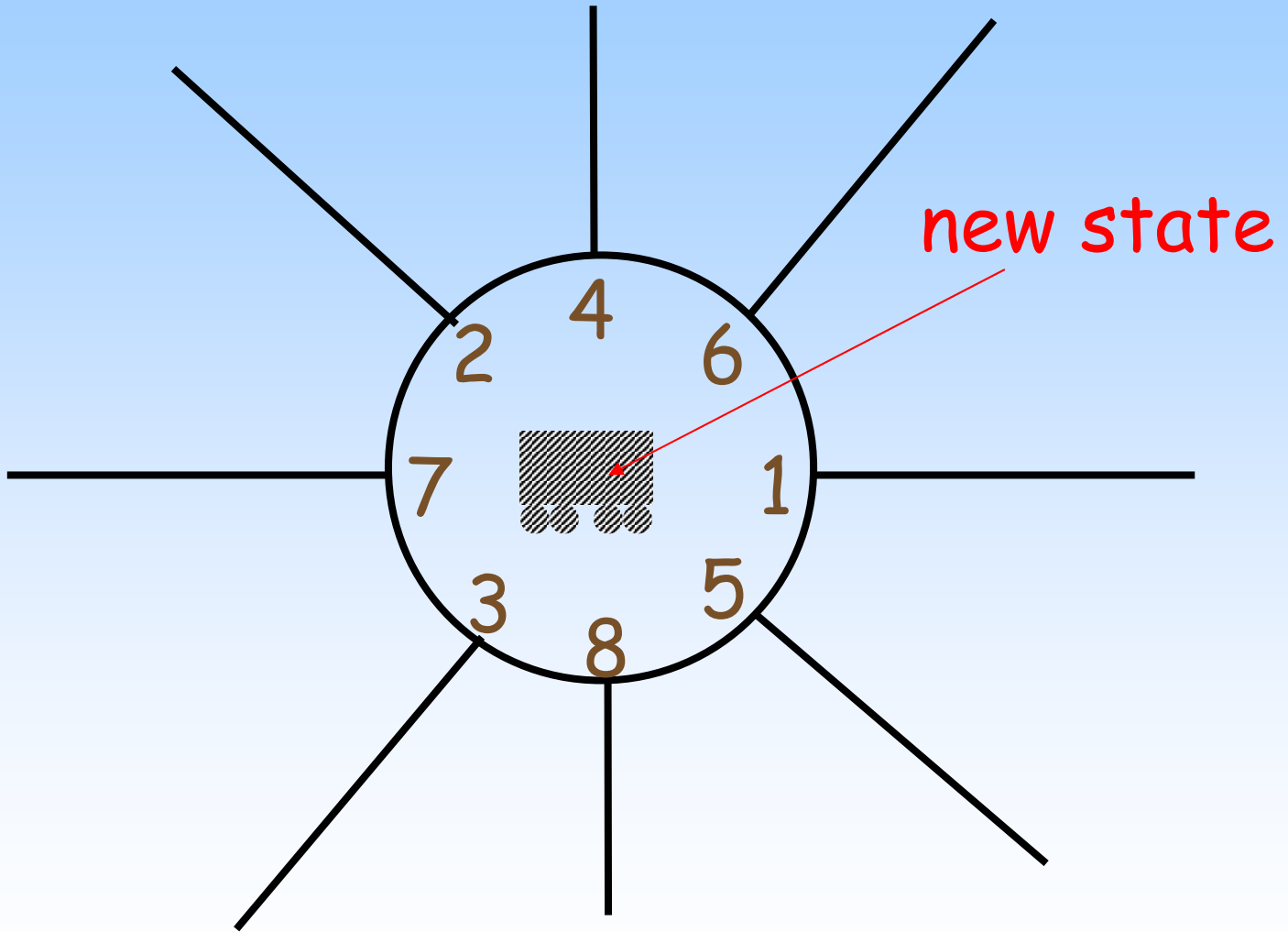
Robot's actions



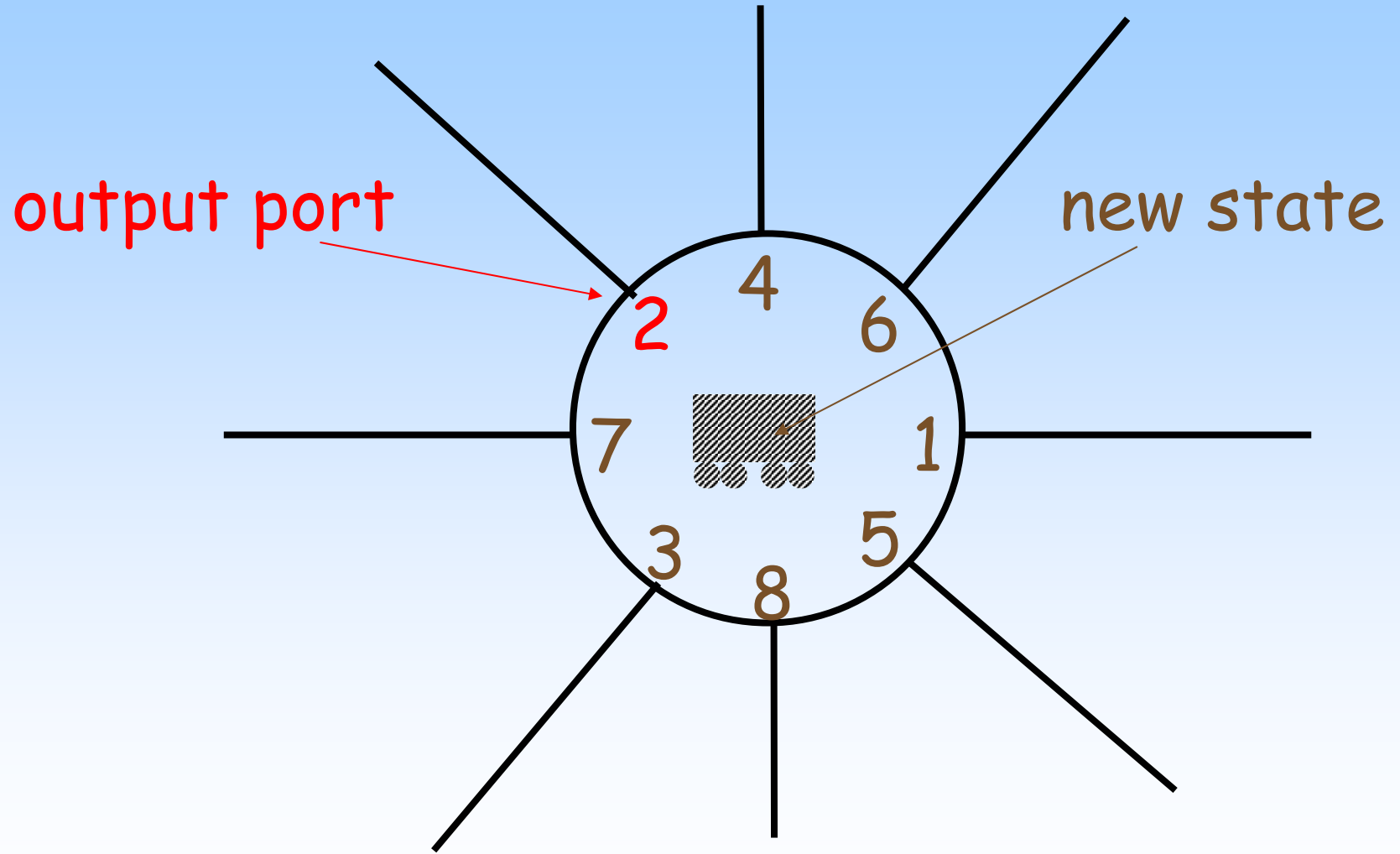
Robot's actions



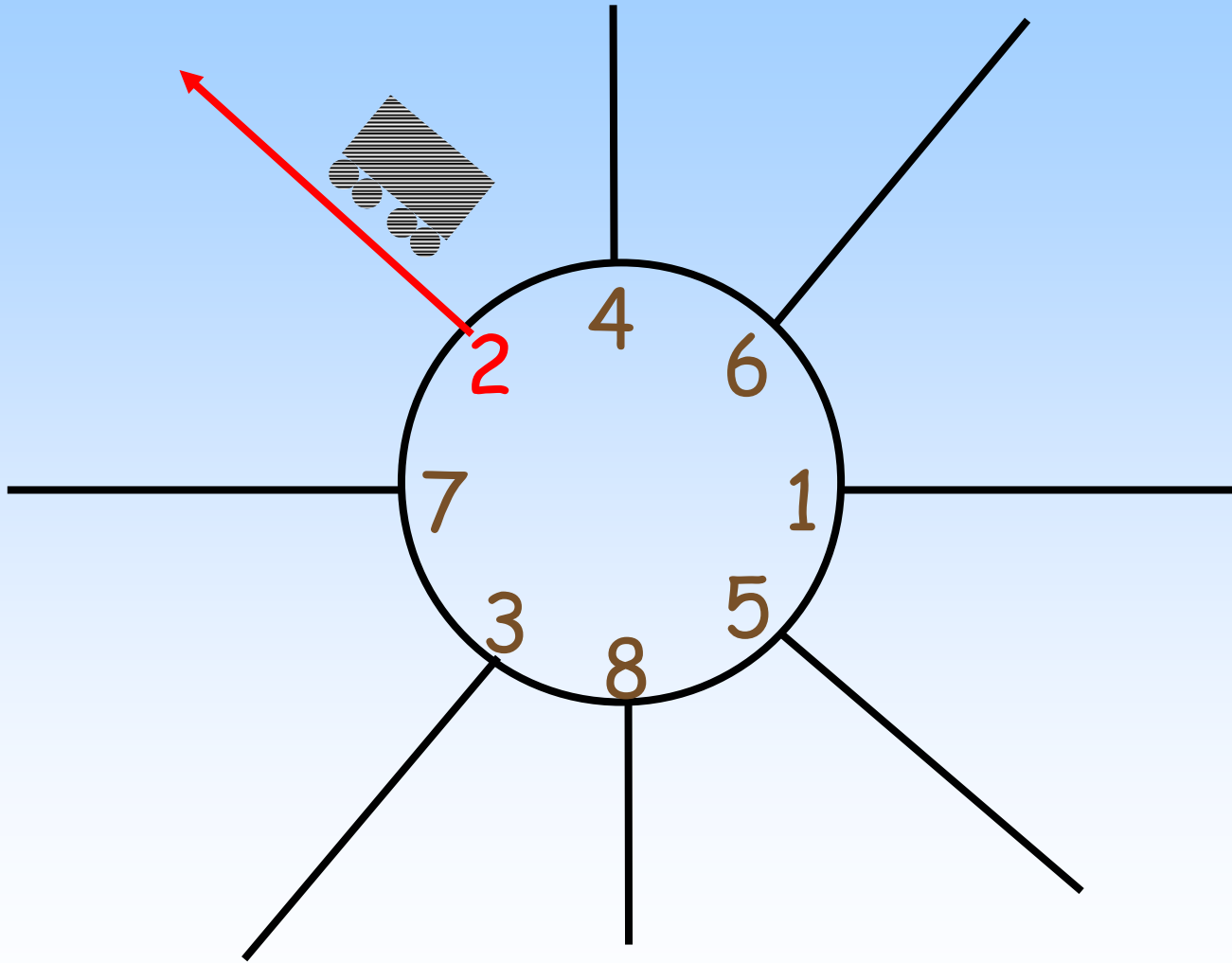
Robot's actions



Robot's actions



Robot's actions



Two different frameworks

Two different frameworks

- Geometric setting
 - rooms with obstacles

Two different frameworks

- Geometric setting
 - rooms with obstacles
- Network exploration
 - graphs

Geometric setting

- Bar-Eli, Berman, Fiat, Yan [JALG 1994]
On-line navigation in a room
- Blum, Raghavan, Schieber [SIAMJC 1997]
Navigating in unfamiliar geometric terrain
- Deng, Kameda, Papadimitriou [JACM 1998]
How to learn an unknown environment I:
the rectilinear case

Graph setting

Graph setting

Exploration **with stop**

a mobile entity in an unknown graph has to

- **traverse** all edges
- **stop** after completing exploration

Graph setting

Exploration **with stop**

a mobile entity in an unknown graph has to

- **traverse** all edges
- **stop** after completing exploration

- Weaker task

perpetual exploration

- the mobile entity is not required to stop

Graph setting

Exploration **with stop**

a mobile entity in an unknown graph has to

- **traverse** all edges
- **stop** after completing exploration

- Weaker task

perpetual exploration

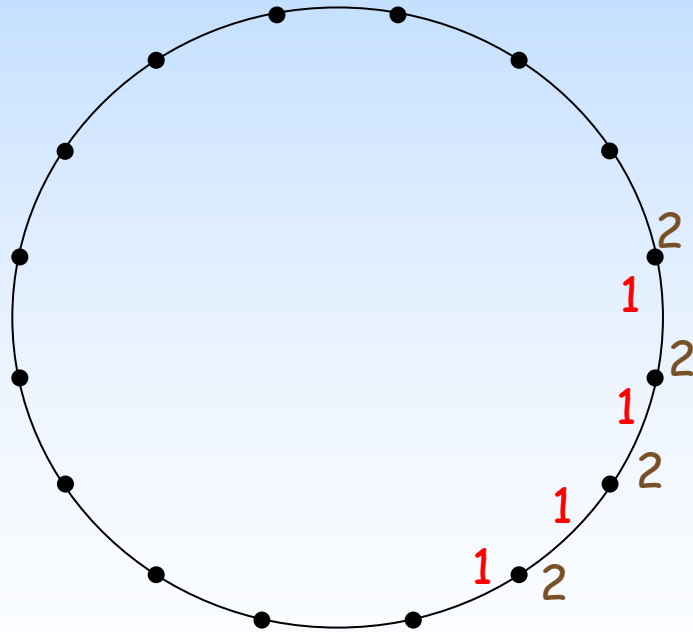
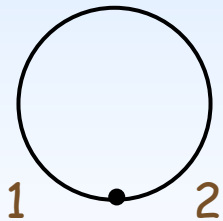
- the mobile entity is not required to stop

- Stronger Task

map drawing

- the mobile entity has to output an edge-labeled isomorphic copy of the graph

Stop \Rightarrow Need of a pebble!



Survey graph exploration

Objective one: minimizing time

- Dudek, Jenkins, Milius, Wilkes [IEEE Transaction on Robotics and Automation 1994]
Robotic exploration as graph construction
- Betke, Rivest, Singh [Machine Learning 1995]
Piecemeal learning of an unknown environment
- Awerbuch, Betke, Rivest, Singh [Information and Computation 1999]
Piecemeal graph exploration by a mobile robot
- Panaite, Pelc [JALG 1999]
Exploring unknown undirected graphs
- Duncan, Kobourov, Kumar [SODA 2001]
Optimal constrained graph exploration

Survey graph exploration

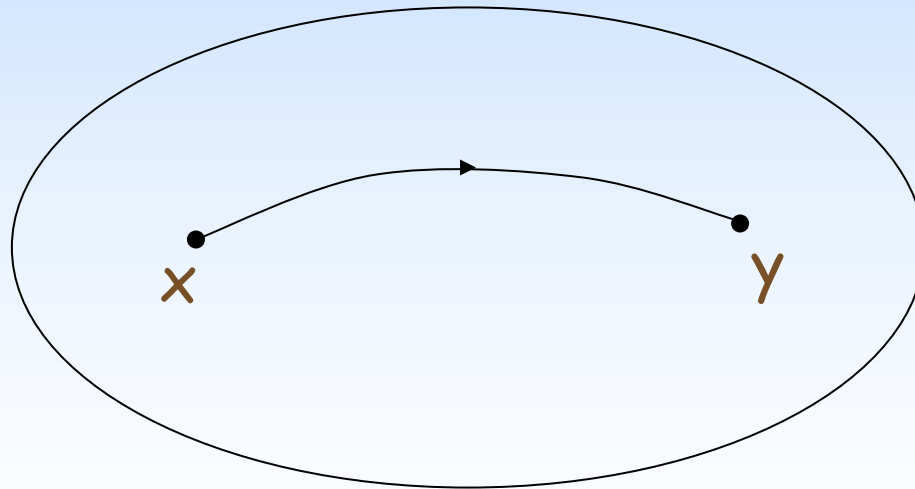
Objective two: minimizing memory

- Budach [Math. Nachrichten 1978]
Automata and labyrinths
- Rollik [Acta Informatica 1980]
Automaten in planaren Graphen
- Diks, Fraigniaud, Kranakis, Pelc
[SODA 2002]
Tree exploration with little memory

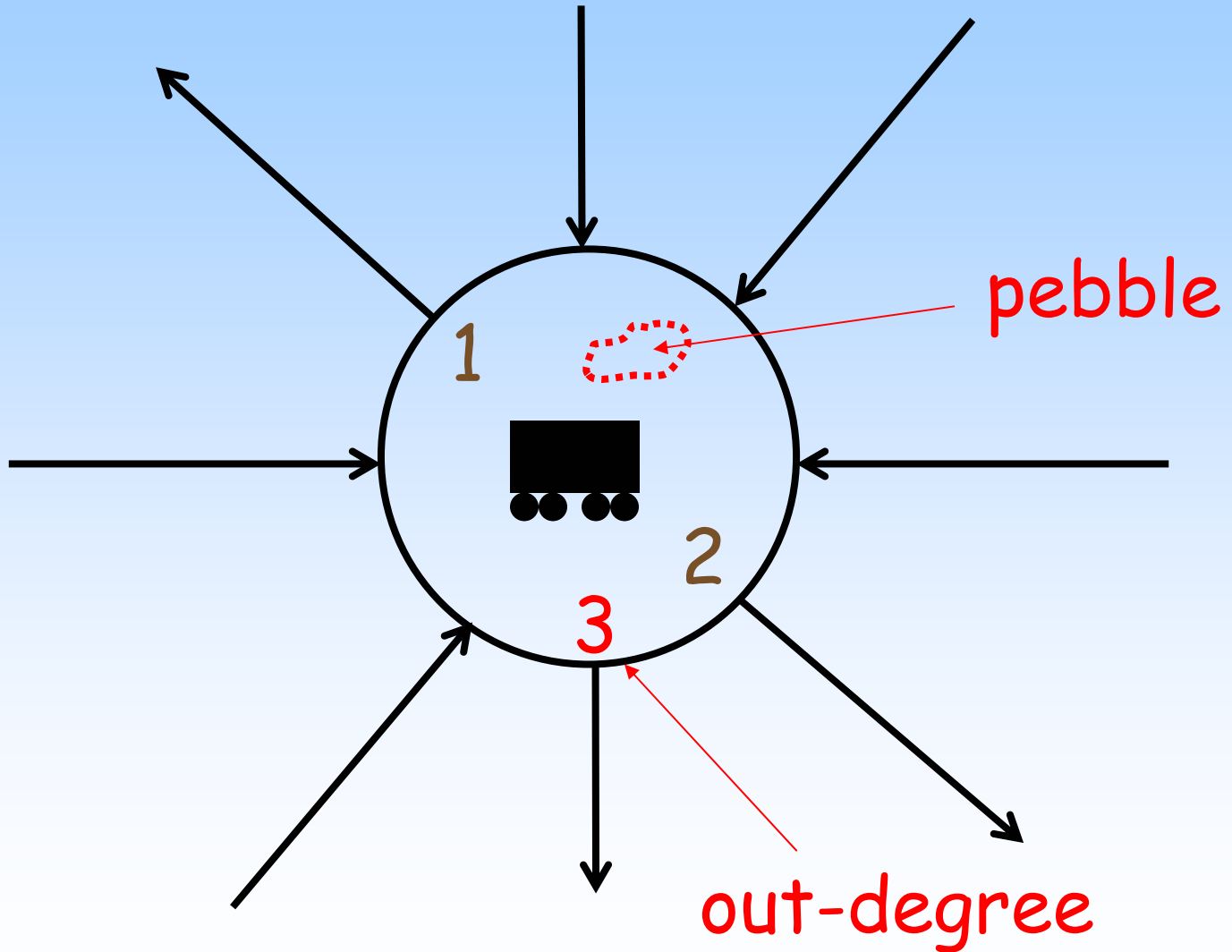
Our framework: Digraphs

Our framework: Digraphs

Difficulty in digraphs: **impossibility to backtrack**



Case of digraphs



Digraph exploration

- Bender, Slonim [FOCS 1994]
The power of team exploration: Two robots can learn unlabeled directed graphs
- Bender, Fernandez, Ron, Sahai, Vadhan [STOC 1998]
The power of a pebble: Exploring and mapping directed graphs
- Deng, Papadimitriou [JGT 1999]
Exploring an unknown graph
- Albers, Henzinger [SICOMP 2000]
Exploring unknown environments

Our results

Exploration with stop in anonymous digraphs

- Lower bound
 - $\Omega(n \log d)$ bits
- Upper bounds
 - compact memory
 - $O(nd \log n)$ bits, one pebble
 - polynomial time
 - $O(n^2d \log n)$ bits, $\Theta(\log \log n)$ pebbles

Lower bound: $\Omega(n \log d)$

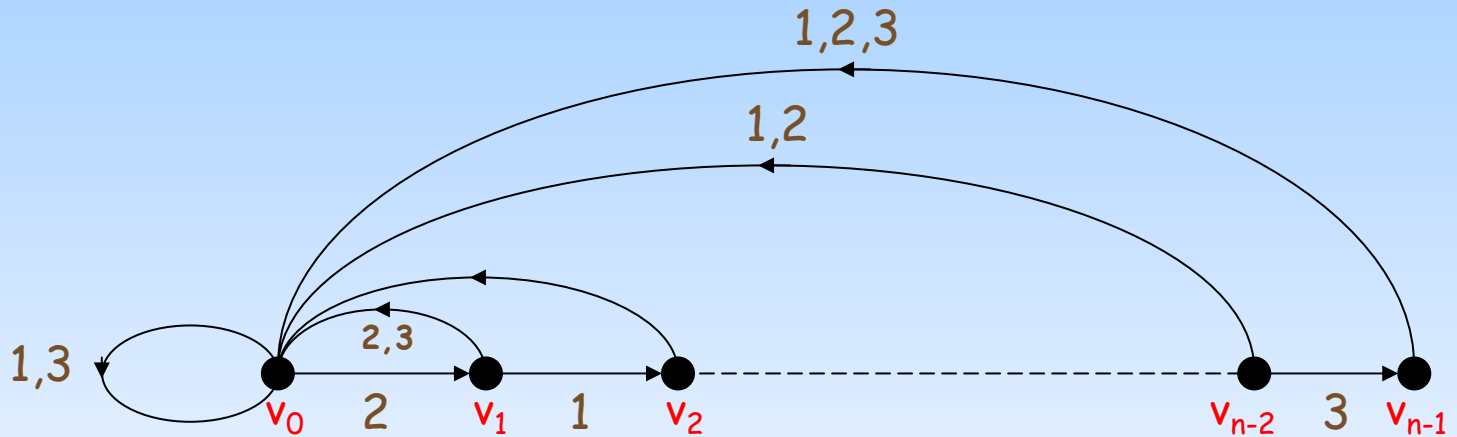
Theorem 1

Exploration cannot be done with less than $\Omega(n \log d)$ bits

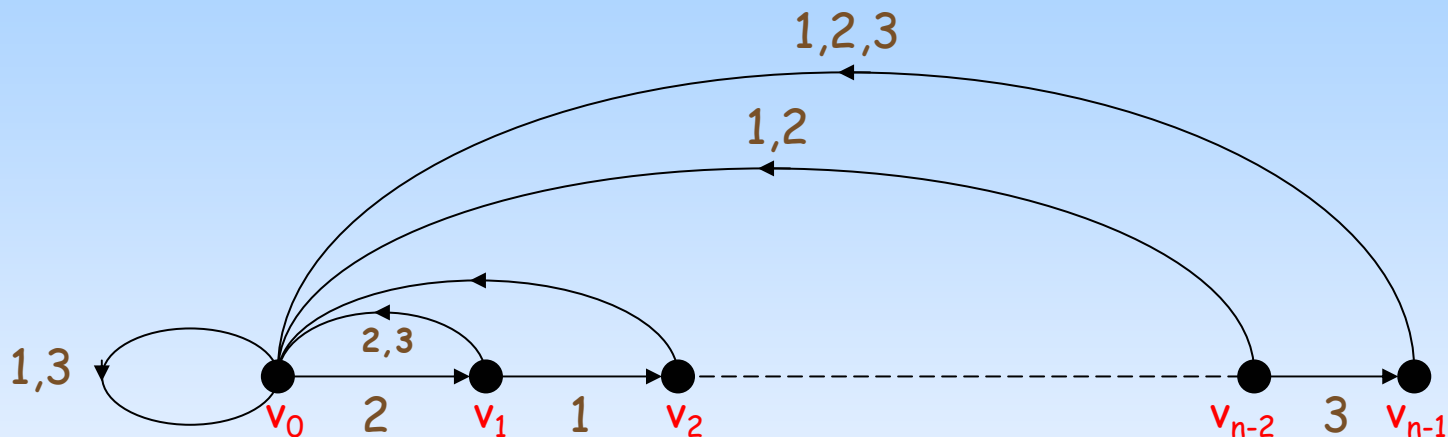
- Additional properties
 - holds even with a linear number of pebbles
 - also holds for perpetual exploration

Proof of $\Omega(n \log d)$

Proof of $\Omega(n \log d)$



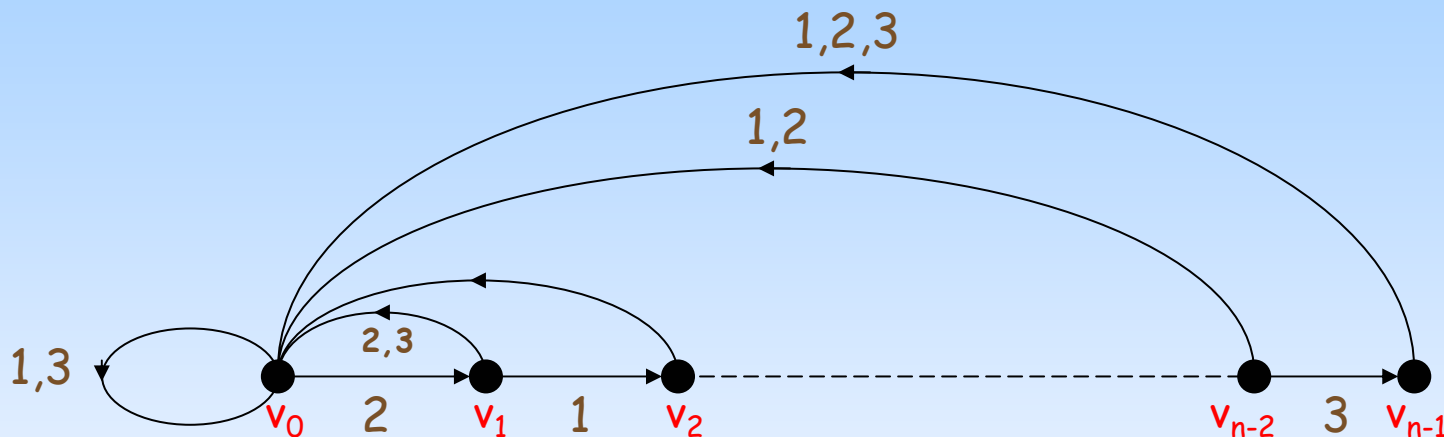
Proof of $\Omega(n \log d)$



Number of *system's states* before a full run

- number of *robot's states*: 2^k (k bits of memory)
- number of *configurations* for the pebbles: 2^n

Proof of $\Omega(n \log d)$

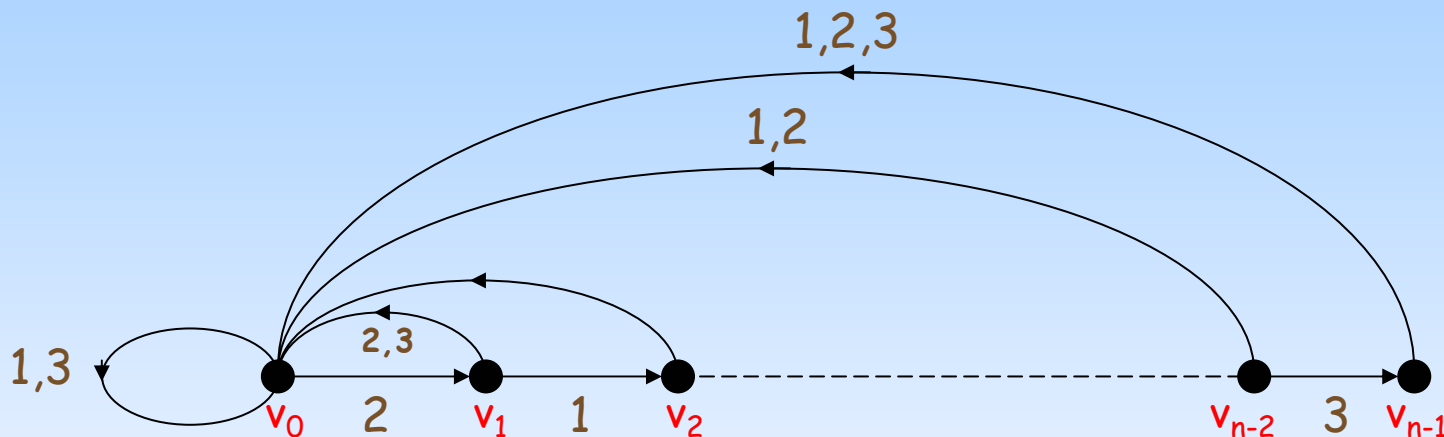


Number of *system's states* before a full run

- number of robot's states: 2^k (k bits of memory)
- number of configurations for the pebbles: 2^n

Number of combination locks: d^{n-1}

Proof of $\Omega(n \log d)$



Number of **system's states** before a full run

- number of robot's states: 2^k (k bits of memory)
- number of configurations for the pebbles: 2^n

Number of combination locks: d^{n-1}

We have $2^k \times 2^n \geq d^{n-1} \rightarrow k \geq n \log(d/2)$

Upper bound: $O(nd \log n)$

Theorem 2

Algorithm **Test-all-maps** accomplishes exploration with a robot

- of $O(nd \log n)$ bits
- using **one pebble**

- **Remarks**

- accomplishes map drawing
- time complexity: exponential

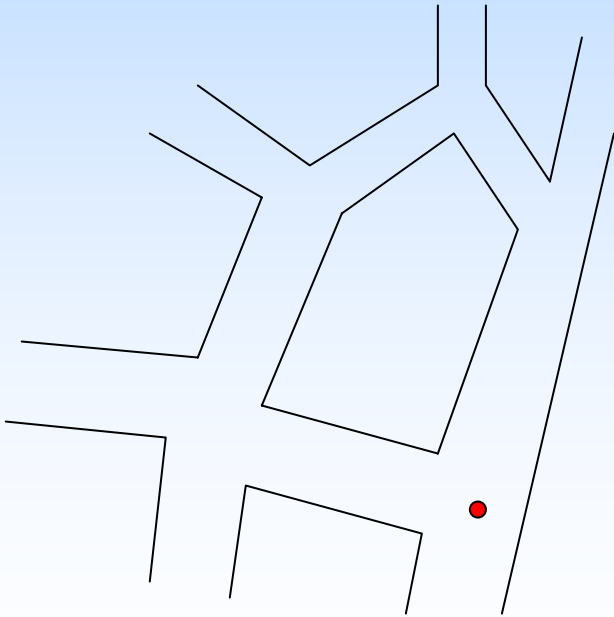
Description of test-all-maps

Description of test-all-maps

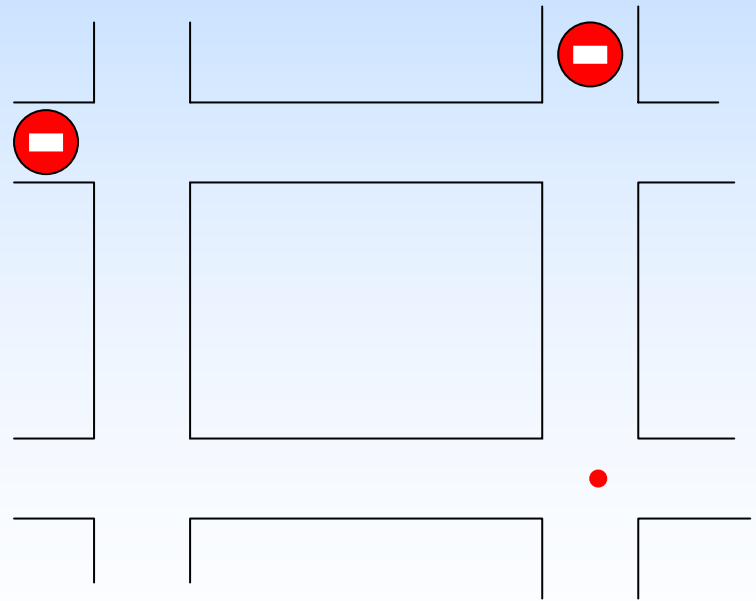
- Main idea
 - the robot tests **all the maps**

Description of test-all-maps

- Main idea
 - the robot tests **all the maps**



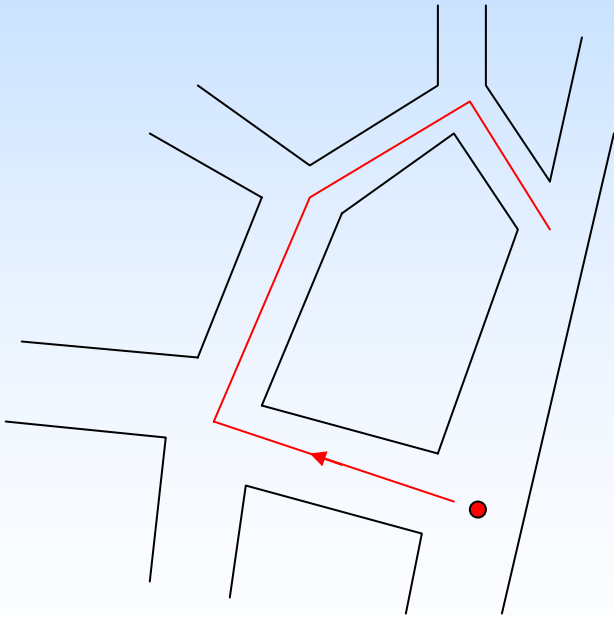
The real city (Paris)



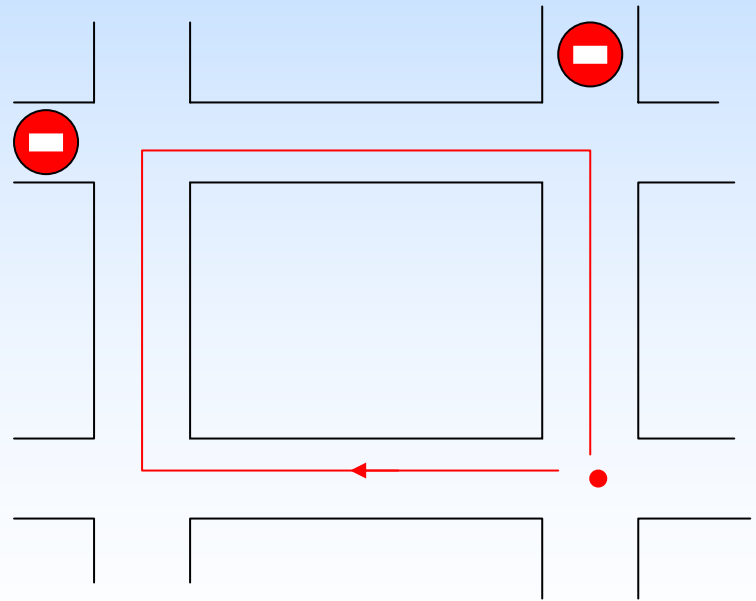
A candidate map (New York)

Description of test-all-maps

- Main idea
 - the robot tests **all the maps**



The real city (Paris)



A candidate map (New York)

Upper bound: $O(n^2d \log n)$

- Theorem 3

Algorithm **Compacted-Explore-and-map** accomplishes exploration with a robot

- of $O(n^2d \log n)$ bits
- using $\Theta(\log \log n)$ pebbles

- Remarks

- optimized variant of the algorithm of **Bender et al.** [STOC 1998]
- does map drawing
- time complexity: **polynomial**

Conclusion

- Results

- Lower bound

- $\Omega(n \log d)$ bits

- Upper bounds

- $O(nd \log n)$ bits, one pebble

- $O(n^2d \log n)$ bits, $\Theta(\log \log n)$ pebbles