Part 2

An introduction to mobile robotics seen through the eyes of distributed computing

Eric Goubault

Bernardo Hummes

June 15, 2022

Outline

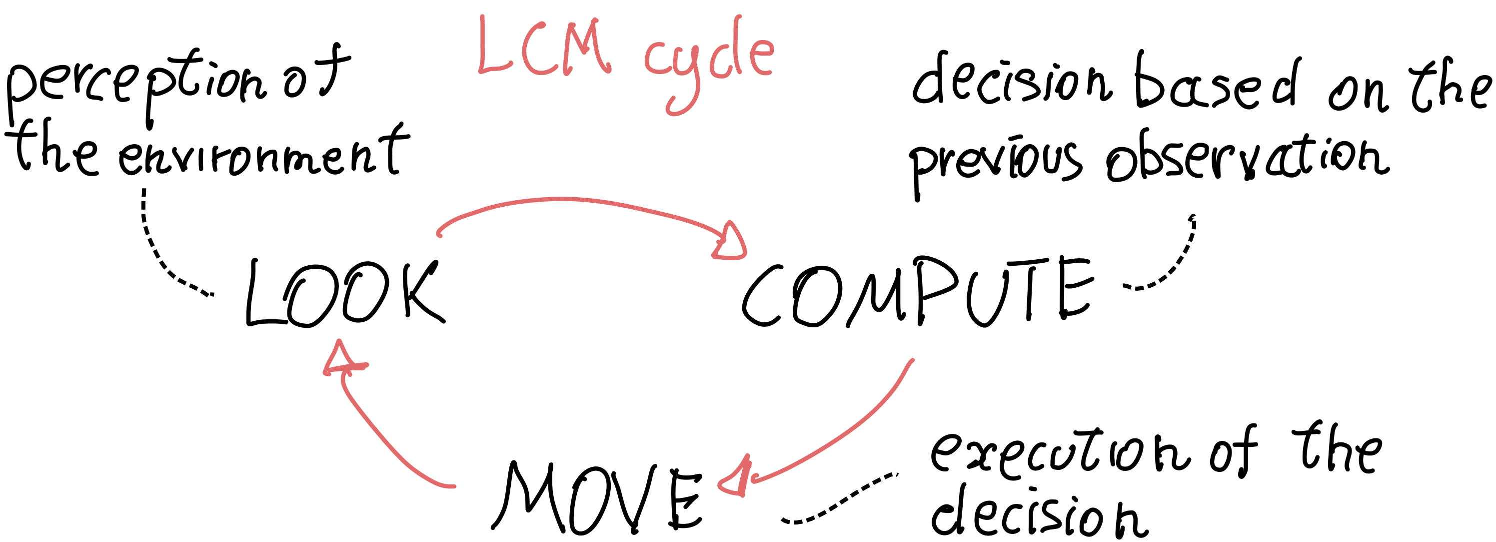
- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) curent results
- Robot tasks
- Exploration as a task

Outline

- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) current results
- Robot tasks
- Exploration as a task

LOOK-COMPUTE-MOVE

- Theoretical framework for studying different robotic scenarios in an unified view



LOOK-COMPUTE-MOVE

Agents

- Movement

  random or deterministic
- Identity

  labeled or anonymous
- Knowledge

  map and other agents
- Communication

  time and distance

* among many other classifiations

LOOK - COMPUTE - MOVE

Agents

- Movement
  random or deterministic
- Identity
  labeled or anonymous
- Knowledge
  map and other agents
- Communication
  time and distance

Environment

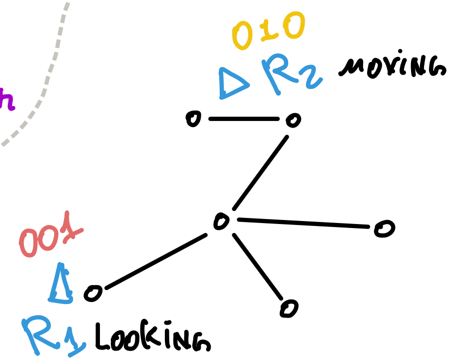
- World
  graph or euclidean space
- Labeling
  local or global / sense of direction
- Time
  synchonous or asynchonous
- Storage
  tokens and messages

* among many other classifiations

EXTENDED ASYNCHRONOUS LUMINOUS ROBOTS (EALR)

stop moving preserving light

LCM model with anonymous robots capable of failing in a known graph that move assynchronously with identical deterministic algorithms equiped with lights for implicit communication with arbitrary starting times



they keep ⊥ in their light until activation

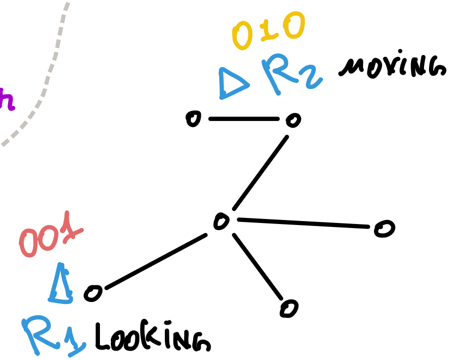
any information can be encoded, needs to be seen to transmit data

no common clock, actions happen at unpredictable moments

EXTENDED ASYNCHRONOUS LUMINOUS ROBOTS (EALR)

stop moving preserving light

LCM model with anonymous robots capable of failing in a known graph that move assynchronously with identical determinstic algorithms equiped with lights for implicit communication with arbitrary starting times



they keep ⊥ in their light until activation

any information can be encoded, needs to be seen to transmit data

no common clock, actions happen at unpredictable moments

EXTENDED ASYNCHRONOUS LUMINOUS ROBOTS (EALR)

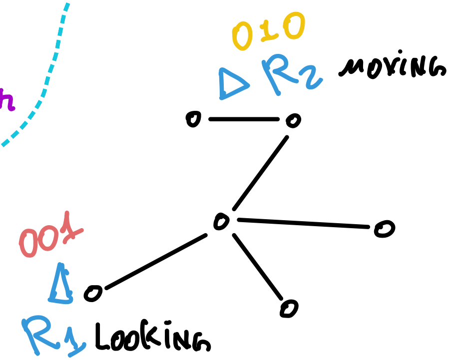
LCM model with anonymous robots capable of failing in a known graph that move assynchronously with identical deterministic algorithms equiped with lights for implicit communication with arbitrary starting times

stop moving preserving light

they keep ⊥ in their light until activation

any information can be encoded, needs to be seen to transmit data

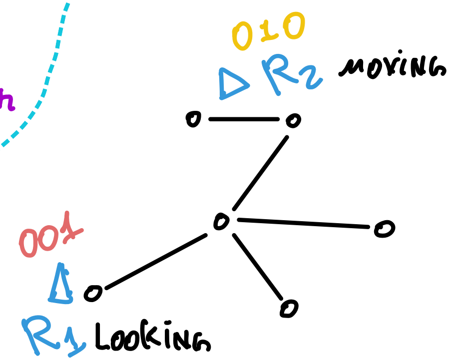
no common clock, actions happen at unpredictable moments



EXTENDED ASYNCHRONOUS LUMINOUS ROBOTS (EALR)

stop moving preserving light

LCM model with anonymous robots capable of failing in a known graph that move assynchronously with identical determinstic algorithms equiped with lights for implicit communication with arbitrary starting times



they keep ⊥ in their light until activation

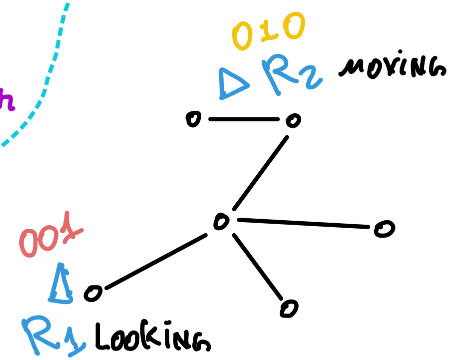
any information can be encoded, needs to be seen to transmit data

no common clock, actions happen at unpredictable moments

EXTENDED ASYNCHRONOUS LUMINOUS ROBOTS (EALR)

stop moving preserving light

LCM model with anonymous robots capable of failing in a known graph that move assynchronously with identical deterministic algorithms equiped with lights for implicit communication with arbitrary starting times



they keep $\perp$ in their light until activation

any information can be encoded, needs to be seen to transmit data

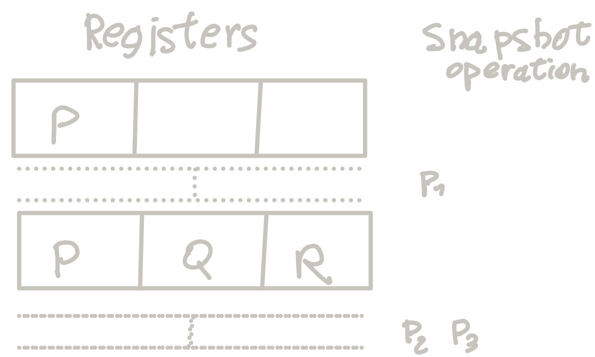
no common clock, actions happen at unpredictable moments

WAIT FREE SHARED MEMORY (WFSM)

Distributed computing model where undistinguishable programs that can fail assynchronously execute atomic operations of SNAPSHOT, COMPUTE and WRITE in a shared memory space with arbitrary starting times

P_1	P_2	P_3
Write		
Snap		
	Write	Write
	Snap	Snap
{P}	{P,Q,R}	{P,Q,R}

they cannot wait for others to finish their operations



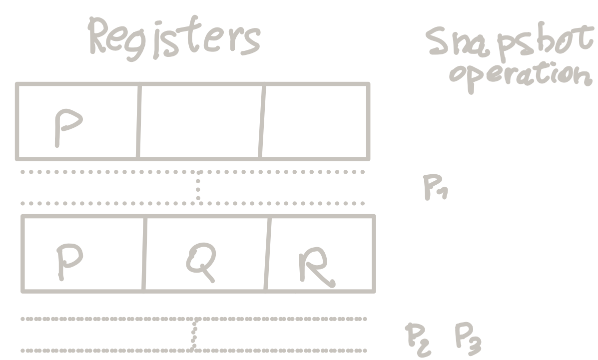
a program writes to a single register but reads all of them

WAIT FREE SHARED MEMORY (WFSM)

Distributed computing model where undistinguishable programs that can fail assynchronously execute atomic operations of SNAPSHOT, COMPUTE and WRITE in a shared memory space with arbitrary starting times

→ they cannot wait for others to finish their operations

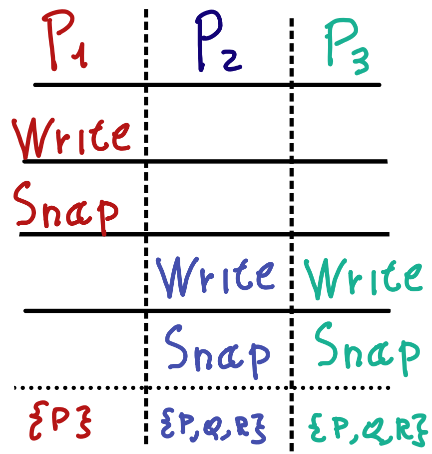
P_1	P_2	P_3
Write		
Snap		
	Write	Write
	Snap	Snap
{P}	{P,Q,R}	{P,Q,R}



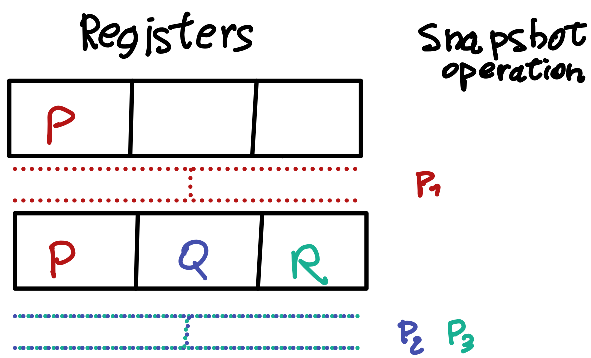
a program writes to a single register but reads all of them

WAIT FREE SHARED MEMORY (WFSM)

Distributed computing model where undistinguishable programs that can fail assynchronously execute atomic operations of SNAPSHOT, COMPUTE and WRITE in a shared memory space with arbitrary starting times



they cannot wait for others to finish their operations



a program writes to a single register but reads all of them

Outline

- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) curent results
- Robot tasks
- Exploration as a task

EQUIVALENCE BETWEEN EALR AND WFSM

Theorem:

A robot task $\mathcal{T}$ on a graph is solvable in the EALR model with $N \geq 2$ robots tolerating f failures if and only if $\mathcal{T}$ is solvable in the WFSM model with $N \geq 2$ processes tolerating up to f failures.

---

* From Alcantara et al, The topology of look-compute-move robot wait-free algorithms with hard termination.

SIMULATING EALR IN WFSM

global view of the graph with the positions and lights, $\perp$ lights are kept invisible

global view of all registers and their contents, $\perp$ is writen for inactive robots

LOOK $\rightarrow$ SNAPSHOT

COMPUTE $\rightarrow$ COMPUTE

MOVE (V_i, C_i) $\rightarrow$ WRITE$((V_i, C_i))$

robot i moves to the vertex V_i and sets the color C_i

process writes $((V_i, C_i))$ to the memory at its register

Equivalence between EALR and WFSM

SIMULATING EALR IN WFSM

global view of the graph with the positions and lights, $\perp$ lights are kept invisible

global view of all registers and their contents, $\perp$ is writen for inactive robots

LOOK $\rightarrow$ SNAPSHOT

COMPUTE $\rightarrow$ COMPUTE

MOVE (V_i, C_i) $\rightarrow$ WRITE$((V_i, C_i))$

robot i moves to the vertex V_i and sets the color C_i

process writes $((V_i, C_i))$ to the memory at its register

SIMULATING EALR IN WFSM

global view of the graph with the positions and lights, $\perp$ lights are kept invisible

global view of all registers and their contents, $\perp$ is writen for inactive robots

LOOK $\rightarrow$ SNAPSHOT

COMPUTE $\rightarrow$ COMPUTE

MOVE (V_i, C_i) $\rightarrow$ WRITE$(\langle V_i, C_i \rangle)$

robot i moves to the vertex V_i and sets the color C_i

process writes $\langle V_i, C_i \rangle$ to the memory at its register

SIMULATING WFSM IN EALR

global view of the graph with the positions and lights, inactive processes have $\perp$ in their lights

global view of all registers and their contents

LOOK $\leftarrow$ SNAPSHOT

COMPUTE $\leftarrow$ COMPUTE

MOVE(–, X) $\leftarrow$ WRITE(X)

robot i moves to whatever adjacent vertex and sets its color to X

process writes X value to the memory at its register

SIMULATING WFSM IN EALR

global view of the graph with the positions and lights, inactive processes have $\perp$ in their lights

global view of all registers and their contents

LOOK $\leftarrow$ SNAPSHOT

COMPUTE $\leftarrow$ COMPUTE

MOVE(–, X) $\leftarrow$ WRITE(X)

robot i moves to whatever adjacent vertex and sets its color to X

process writes X value to the memory at its register

SIMULATING WFSM IN EALR

global view of the graph with the positions and lights, inactive processes have ⊥ in their lights

global view of all registers and their contents

LOOK ← SNAPSHOT

COMPUTE ← COMPUTE

MOVE(–, x) ← WRITE(x)

robot i moves to whatever adjacent vertex and sets its color to X

process writes X value to the memory at its register

Outline

- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) curent results
- Robot tasks
- Exploration as a task

BINARY AND APPROXIMATE CONSENSUS PROBLEM

We wish for $N \geq 2$ nameless processes, with private inputs among a set of possible ones, to decide on a value, respecting the following properties:

- Termination: every correct process decides a value.
- Validity: the decided value must be proposed by at least one process.
- Agreement: all decided values are the same.

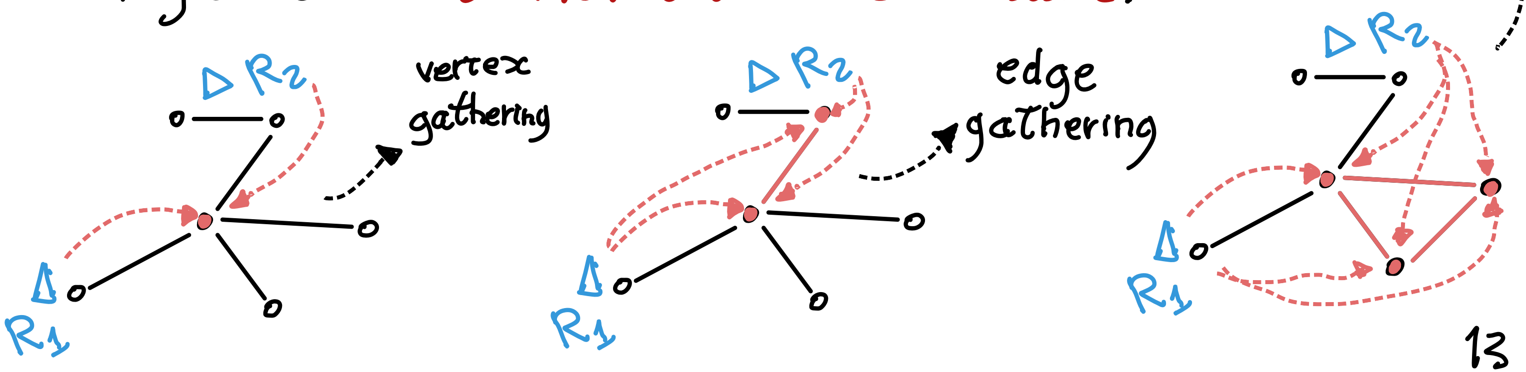
0	1	0	→	1	1	1	→ binary consensus
21	2	0	→	0	2	2	→ K-set agreement (K=2)

VERTEX AND APPROXIMATE GATHERING PROBLEM

We wish for $N \geq 2$ undistinguishable robots executing identical algorithms to decide on a vertex to move to, respecting the following properties:

- Termination: every robot decides a vertex in a bounded number of LCM cycles.
- Validity: the decided vertex cannot be fixed in advance.
- Agreement: all decided vertices are the same.

1-gathering



Impossibility of Gathering with Termination

Theorem

The gathering problem with termination is unsolvable by any algorithm in the ALR model even if robots have powerful capabilities, namely, they are non-oblivious and able to detect multiplicities, share the same labeling of G and have an unbounded number of lights.

Impossibility of Gathering with Termination

Idea of the proof

Algorithm A in EALR that respects termination and validity while solving gathering.

Algorithm B in WFSM that solves binary consensus can be built using simulated A.

Binary consensus is knowingly impossible.

Algorithm A cannot exist.

Gathering with termination is impossible

*main reason is the validity property

IMPOSSIBILITY OF GATHERING WITH TERMINATION

Idea of the proof

Algorithm A in EALR that respects termination and validity while solving gathering.

Algorithm B in WFSM that solves binary consensus can be built using simulated A.

Binary consensus is knowingly impossible.

Algorithm A cannot exist.

Gathering with termination is impossible

*main reason is the validity property

IMPOSSIBILITY OF GATHERING WITH TERMINATION

Idea of the proof

Algorithm A in EALR that respects termination and validity while solving gathering.

Algorithm B in WFSM that solves binary consensus can be built using simulated A.

Binary consensus is knowingly impossible.

Algorithm A cannot exist.

Gathering with termination is impossible

*main reason is the validity property

IMPOSSIBILITY OF GATHERING WITH TERMINATION

Idea of the proof

Algorithm A in EALR that respects termination and validity while solving gathering.

Algorithm B in WFSM that solves binary consensus can be built using simulated A.

Binary consensus is knowingly impossible.

Algorithm A cannot exist.

Gathering with termination is impossible

*main reason is the validity property

IMPOSSIBILITY OF GATHERING WITH TERMINATION

Idea of the proof

Algorithm A in EALR that respects termination and validity while solving gathering.

Algorithm B in WFSM that solves binary consensus can be built using simulated A.

Binary consensus is knowingly impossible.

Algorithm A cannot exist.

Gathering with termination is impossible

*main reason is the validity property

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Theorem

Let G be a connected cyclic graph. There is no algorithm that solves edge gathering on G for $N \geq 3$ robots tolerating two failures in the STRONG version of the EALR model.

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Idea of the proof

at most k different values are decided

Algorithm A in EALR that respects termination and validity while solving edge-gathering on a cyclic graph

Algorithm B in WFSM that solves 2-set agreement for 3 processes and 2 failures using simulated A.

Algorithm C in WFSM that is similar to B, but with $N \geq 3$ processes using B via BG simulation

2-Set agreement is impossible to solve → Algorithm A cannot exist → Edge gathering on cyclic graphs is impossible

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Idea of the proof

at most k different values are decided

Algorithm A in EALR that respects termination and validity while solving edge-gathering on a cyclic graph

Algorithm B in WFSM that solves 2-set agreement for 3 processes and 2 failures using simulated A.

Algorithm C in WFSM that is similar to B, but with N≥3 processes using B via BG simulation

2-Set agreement is impossible to solve

Algorithm A cannot exist

Edge gathering on cyclic graphs is impossible

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Idea of the proof

at most k different values are decided

Algorithm A in EALR that respects termination and validity while solving edge-gathering on a cyclic graph

Algorithm B in WFSM that solves 2-set agreement for 3 processes and 2 failures using simulated A.

Algorithm C in WFSM that is similar to B, but with $N \geq 3$ processes using B via BG simulation

2-Set agreement is impossible to solve → Algorithm A cannot exist → Edge gathering on cyclic graphs is impossible

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Idea of the proof

at most k different values are decided

Algorithm A in EALR that respects termination and validity while solving edge-gathering on a cyclic graph

Algorithm B in WFSM that solves 2-set agreement for 3 processes and 2 failures using simulated A.

Algorithm C in WFSM that is similar to B, but with $N \geq 3$ processes using B via BG simulation

2-Set agreement is impossible to solve

Algorithm A cannot exist

Edge gathering on cyclic graphs is impossible

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Idea of the proof

at most k different values are decided

Algorithm A in EALR that respects termination and validity while solving edge-gathering on a cyclic graph

Algorithm B in WFSM that solves 2-set agreement for 3 processes and 2 failures using simulated A.

Algorithm C in WFSM that is similar to B, but with N≥3 processes using B via BG simulation

2-Set agreement is impossible to solve

Algorithm A cannot exist

Edge gathering on cyclic graphs is impossible

IMPOSSIBILITY OF EDGE-GATHERING ON CYCLIC GRAPHS

Idea of the proof

at most k different values are decided

Algorithm A in EALR that respects termination and validity while solving edge-gathering on a cyclic graph

Algorithm B in WFSM that solves 2-set agreement for 3 processes and 2 failures using simulated A.

Algorithm C in WFSM that is similar to B, but with $N \geq 3$ processes using B via BG simulation

2-Set agreement is impossible to solve

Algorithm A cannot exist

$\boxed{\text{Edge gathering on cyclic graphs is impossible}}$

POSSIBILITY OF EDGE-GATHERING ON TREES

Theorem

The edge gathering problem is solvable in the EALR model for $N \geq 2$ robots on any tree T in $diam(T)-1$ rounds using $diam(T)-1$ distinct light colors an tolerating up to $N-1$ crash failures.

POSSIBILITY OF EDGE-GATHERING ON TREES

extreme case for edge-gathering

Each robot r_i executes the same algorithm for diam(T) -1 rounds:

the robots count their rounds and show in their lights

1. A spanning tree is constructed of the vertices occupied by robots in the maximal round + r_i

2. r_i will either be at a leaf of the tree or not

2.1 If r_i is in a leaf, it moves deeper into the tree

robots in the maximal round serve as leaders, orienting the others

2.2 Otherwise, r_i stays still

• As their positions are constrained by the initial subtree and they will always get closer, convergence happens up to an edge

POSSIBILITY OF EDGE-GATHERING ON TREES

extreme case for edge-gathering

Each robot r_i executes the same algorithm for diam(T) -1 rounds:

the robots count their rounds and show in their lights

1. A spanning tree is constructed of the vertices occupied by robots in the maximal round + r_i

2. r_i will either be at a leaf of the tree or not

2.1 If r_i is in a leaf, it moves deeper into the tree

robots in the maximal round serve as leaders, orienting the others

2.2 Otherwise, r_i stays still

- As Their positions are constrained by the initial subtree and They will always get closer, convergence happens up to an edge

POSSIBILITY OF EDGE-GATHERING ON TREES

extreme case for edge-gathering

Each robot r_i executes the same algorithm for diam(T) -1 rounds:

the robots count their rounds and show in their lights

1. A spanning tree is constructed of the vertices occupied by robots in the maximal round + r_i

2. r_i will either be at a leaf of the tree or not

2.1 If r_i is in a leaf, it moves deeper into the tree

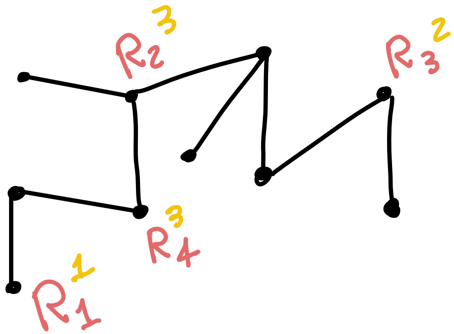
2.2 Otherwise, r_i stays still

robots in the maximal round serve as leaders, orienting the others

- As their positions are constrained by the initial subtree and they will always get closer, convergence happens up to an edge

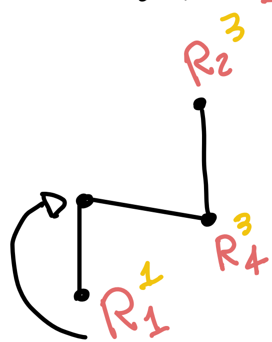
POSSIBILITY OF EDGE-GATHERING ON TREES

T:

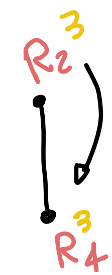


spanning trees from T + direction of movement:

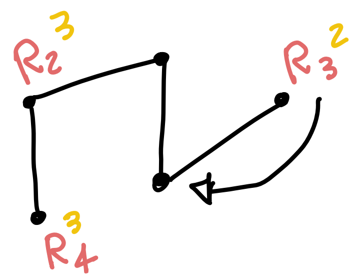
for R_1



for R_2



for R_3



for R_4

*: unless they are past round diam(T) -1

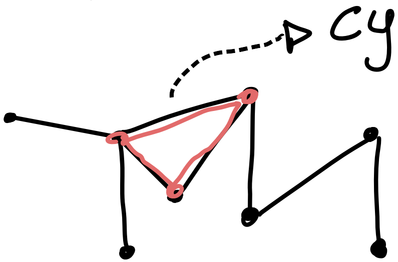
Outline

- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) curent results
- Robot tasks
- Exploration as a task

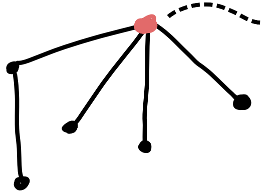
(SOME) RESULTS

Problem	# robots	# lights	graph	solvability
gathering	$N \geq 2$	unbounded	connected	Impossible
edge-gathering	$N \geq 2$	diam(T) - 1	tree	Possible
edge-gathering	$N \geq 3$	0	connected + diam ≥ 3	Impossible
edge-gathering	$N \geq 3$	unbounded	with a cycle	Impossible
1-gathering	$N \geq 2$	0	dominating vertex	Possible
1-gathering	$N \geq 2$	bounded	clique graph is a tree	Possible
1-gathering	$N \geq 3$	unbounded	with cycles + no triangles	Impossible

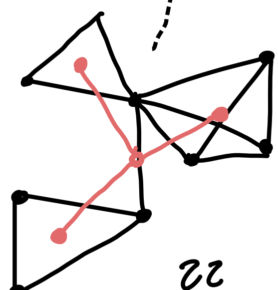
clique graph is a tree



cycle and triangle



dominating vertex in a tree



Outline

- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) curent results
- Robot tasks
- Exploration as a task

ROBOT TASKS

solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that $\delta(\mathrm{Subd}(\mathcal{I})) \subseteq \mathcal{O}$

$$\begin{tikzcd} \mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed, "\mathrm{Subd}(\mathcal{I})"'] & \mathcal{O} \\ \mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O} \end{tikzcd}$$

Input complex all sets of vertices where robots can start

$\mathcal{I}/\mathcal{O}$ carrier map each initial configuration is carried to their acceptable solutions

Output complex all sets of vertices where robots can end to satisfy the objective

Protocol complex final state achieved after cycles of execution

decision simplicial map decided vertex of each robot at the end of execution

ROBOT TASKS

solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that

$$\delta(\mathrm{Subd}(\mathcal{I})) \subseteq \mathcal{O}$$

Input complex
all sets of vertices where robots can start

$$\begin{tikzcd}
\mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed, "\mathrm{Subd}(\mathcal{I})"'] & \mathcal{O} \\
\mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O}
\end{tikzcd}$$

I/O carrier map
each initial configuration is carried to their acceptable solutions

Output complex
all sets of vertices where robots can end to satisfy the objective

Subd($\mathcal{I}$)

Protocol complex
final state achieved after cycles of execution

decision simplicial map
decided vertex of each robot at the end of execution

ROBOT TASKS

solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that $\delta(\mathrm{Subd}(\mathcal{I})) \subseteq \mathcal{O}$

Input complex
all sets of vertices where robots can start

$$\begin{tikzcd} \mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed, "\mathrm{Subd}(\mathcal{I})"'] & \mathcal{O} \\ \mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O} \end{tikzcd}$$

I/O carrier map
each initial configuration is carried to their acceptable solutions

Output complex
all sets of vertices where robots can end to satisfy the objective

Subd($\mathcal{I}$)

Protocol complex
final state achieved after cycles of execution

decision simplicial map
decided vertex of each robot at the end of execution

ROBOT TASKS

Solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that

$$\delta(Subd(\mathcal{I})) \subseteq \mathcal{O}$$

Input complex
all sets of vertices where robots can start

$$\begin{tikzcd}
\mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed] & \mathcal{O} \\
\mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O}
\end{tikzcd}$$

$\mathcal{I}/\mathcal{O}$ carrier map
each initial configuration is carried to their acceptable solutions

Output complex
all sets of vertices where robots can end to satisfy the objective

$Subd(\mathcal{I})$

Protocol complex
final state achieved after cycles of execution

decision simplicial map
decided vertex of each robot at the end of execution

ROBOT TASKS

Solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that

$$\delta(\mathrm{Subd}(\mathcal{I})) \subseteq \mathcal{O}$$

Input complex
all sets of vertices where robots can start

$$\begin{tikzcd} \mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed, "\mathrm{Subd}(\mathcal{I})"'] & \mathcal{O} \\ \mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O} \end{tikzcd}$$

I/O carrier map
each initial configuration is carried to their acceptable solutions

Output complex
all sets of vertices where robots can end to satisfy the objective

Protocol complex
final state achieved after cycles of execution

decision simplicial map
decided vertex of each robot at the end of execution

ROBOT TASKS

Solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that

$$\delta(Subd(\mathcal{I})) \subseteq \mathcal{O}$$

Input complex
all sets of vertices where robots can start

$$\begin{tikzcd} \mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed, "Subd(\mathcal{I})"'] & \mathcal{O} \\ \mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O} \end{tikzcd}$$

I/O carrier map
each initial configuration is carried to their acceptable solutions

Output complex
all sets of vertices where robots can end to satisfy the objective

Protocol complex
final state achieved after cycles of execution

decision simplicial map
decided vertex of each robot at the end of execution

ROBOT TASKS

solutions are downwards inclusive, all subsolutions are also solutions

Robotic missions can be expressed and analyzed using using structures from combinatorial topology. A robot task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ will be solvable if it is possible to find the subdivision and simplicial map bellow, such that $\delta(Subd(\mathcal{I})) \subseteq \mathcal{O}$

Input complex
all sets of vertices where robots can start

$$\begin{tikzcd} \mathcal{I} \arrow[r, "\Delta"] \arrow[d, dashed] & \mathcal{O} \\ \mathcal{P} \arrow[r, dashed, "\delta"] & \mathcal{O} \end{tikzcd}$$

$\mathcal{I}/\mathcal{O}$ carrier map
each initial configuration is carried to their acceptable solutions

Output complex
all sets of vertices where robots can end to satisfy the objective

$Subd(\mathcal{I})$

Protocol complex
final state achieved after cycles of execution

decision simplicial map
decided vertex of each robot at the end of execution

Outline

- Robotics and distributed computing models
- Equivalence between models
- Distributed consensus and gathering
- (Some) curent results
- Robot tasks
- Exploration as a task

MAP EXPLORATION

- Terminating exploration: each vertex is visited by at least one robot, eventually all robots stop.
  - ↳ variations: foraging, pursuit/chase
- Exclusive perpetual exploration: each robot visits each vertex infinetely many times, no two robots traverse the same edge or vertex at the same time.
  - ↳ variations: surveillance, patrolling

Note: not representable in the formalization of robot tasks above!

MAP EXPLORATION

- Terminating exploration: each vertex is visited by at least one robot, eventually all robots stop.
  - ↳ variations: foraging, pursuit/chase

- Exclusive perpetual exploration: each robot visits each vertex infinetely many times, no two robots traverse the same edge or vertex at the same time.
  - ↳ variations: surveillance, patrolling

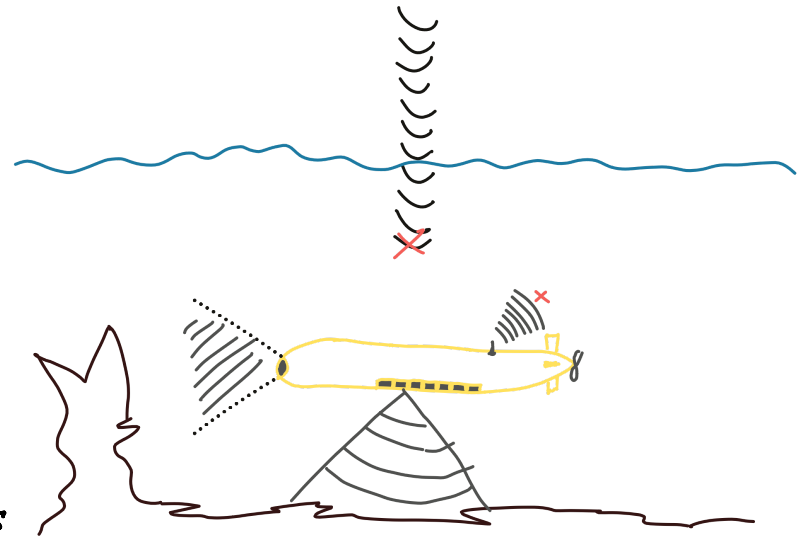
Note: not representable in the formalization of robot tasks above!

REALISTIC SCENARIOS

Together with different types of tasks, it is interesting to explore their possible scenarios, prioritizing realistic ones.

There are several limitations to consider:

- Asynchronous operations
- Limited visibility
- Limited communication
- Faulty movements
- Heterogeneous environments
- Imprecise localization



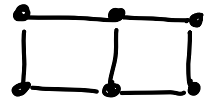
CONCLUSION

- Theoretical frameworks are helpful in dealing with different problems with the same point of view.

- Equivalencies between models are useful for "borrowing" tools, such as the topological approach in robot tasks

- There are multiple formalizations of robotic activities to chose according to context and level of abstraction. It is useful to explore those closer to the applications.

FUTURE WORK



grid graph

- Formalization of exploration tasks and how to extend the current approach for them, relating to underwater exploration.

- In the same context, study variations of the communi-cations model for robots with limited visibility

- Study the usage of grid graphs for the representation of the reachable area of a robot in an homogeneous environment

- Explore the connection with temporal and epistemic modal logics, already studied distributed computing.

Further reading

- The topology of look-compute-move algorithms with hard termination, by Alcántara, Castañeda, Flores-Peñazola and Rajsbaum.

- Brief announcement: variants of approximate agreement on graphs and simplicial complexes, by Ledent.

- Distributed computing by mobile entities: current research in moving and computing, by Flocchini, Prencipe and Santoro.

- Distributed computing through combinatorial topology, by Herlihy, Kozlov and Rajsbaum.

thank you!