Formal Methods for Mobile Robotics

Bernardo Hummes Flores

Eric Goubault   Sylvie Putot   Luc Jaulin

23/02/2023

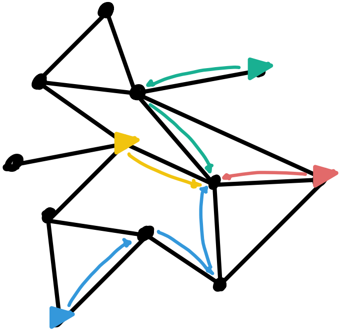
Context

Guaranteed methods for missions with multiple robots without supervision or centralized communication.

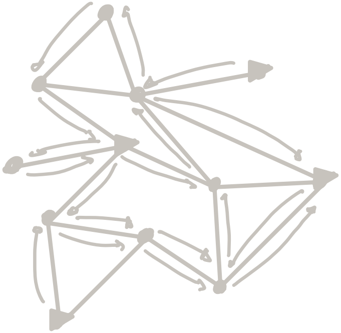
The robots run the same algorithm for a common goal, and They move asynchronously They have limited communication capabilities and are sucseptible To failures.

The environment is discretized in a graph.

Distributed robotics algorithms

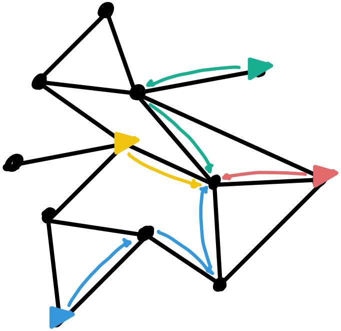


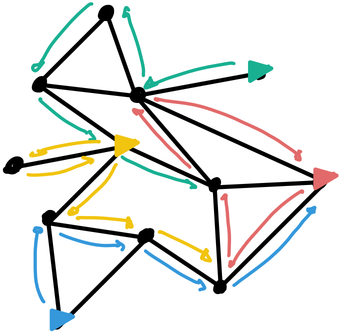
Converging tasks
- gathering
- pattern formation
- ...



Diverging Tasks
- exploration
- patrolling
- ...

Distributed robotics algorithms



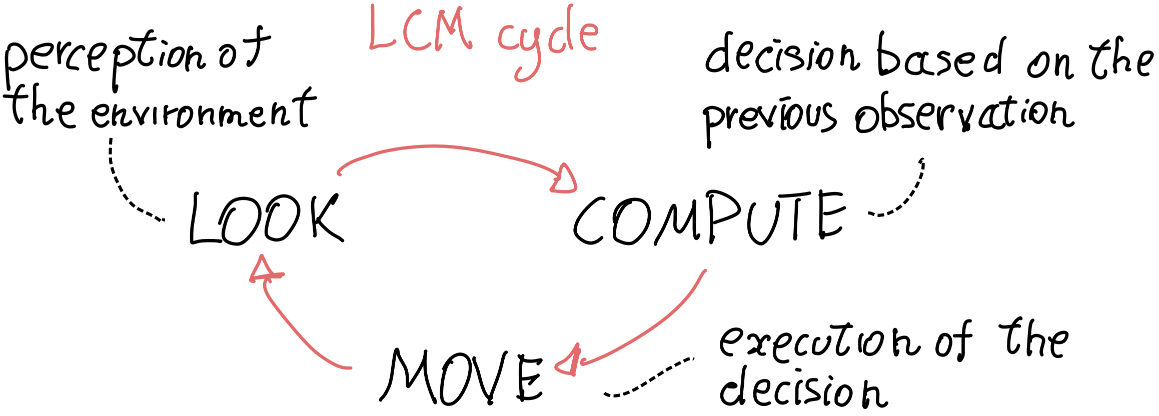


Converging tasks
- gathering
- pattern formation
- ...

Diverging Tasks
- exploration
- patrolling
- ...

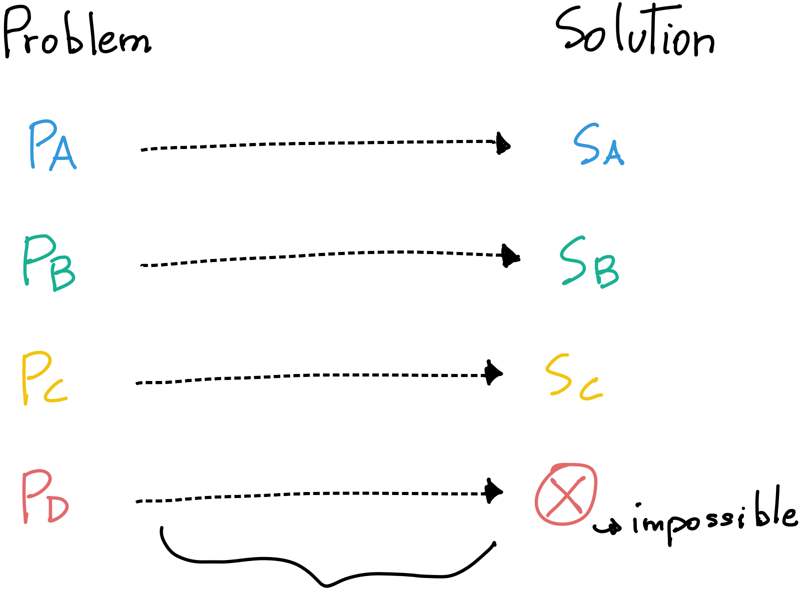
Look-compute-move model

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



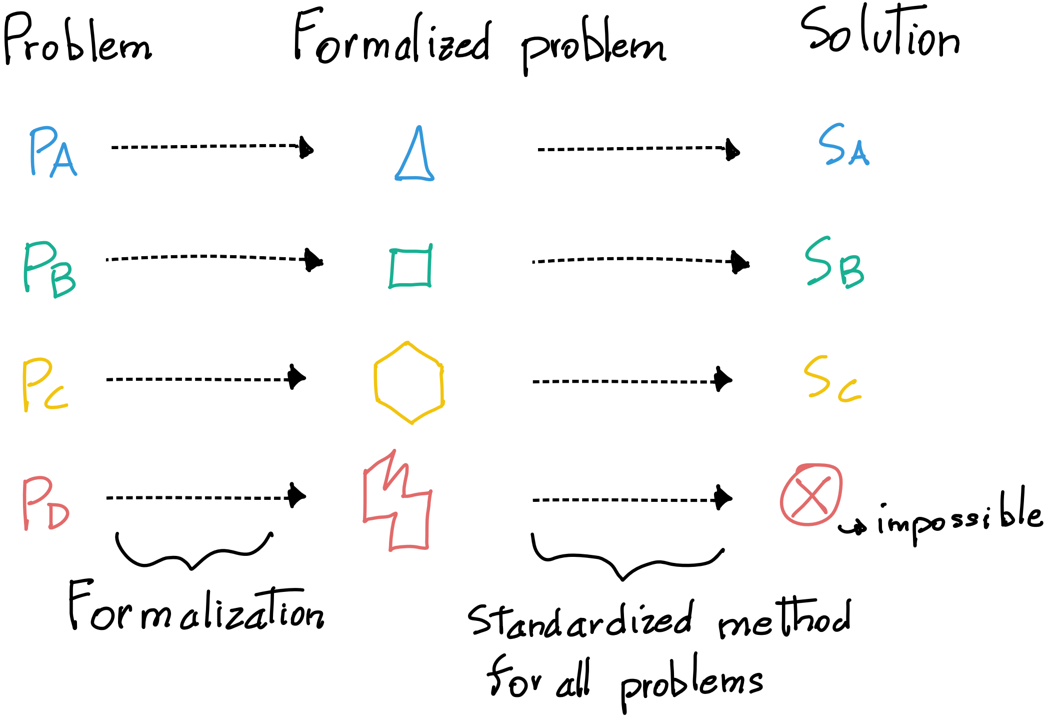
LCM cycle
perception of the environment
LOOK
decision based on the previous observation
COMPUTE
MOVE
execution of the decision

Ad-hoc problem solving

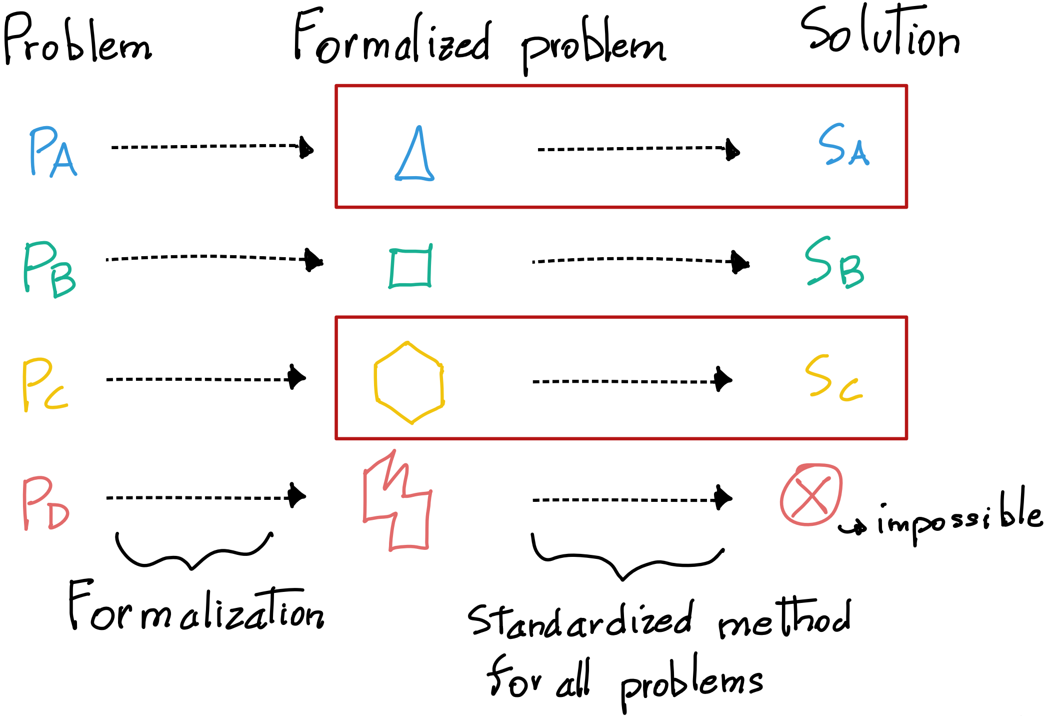


Ad-hoc methods for solving each problem

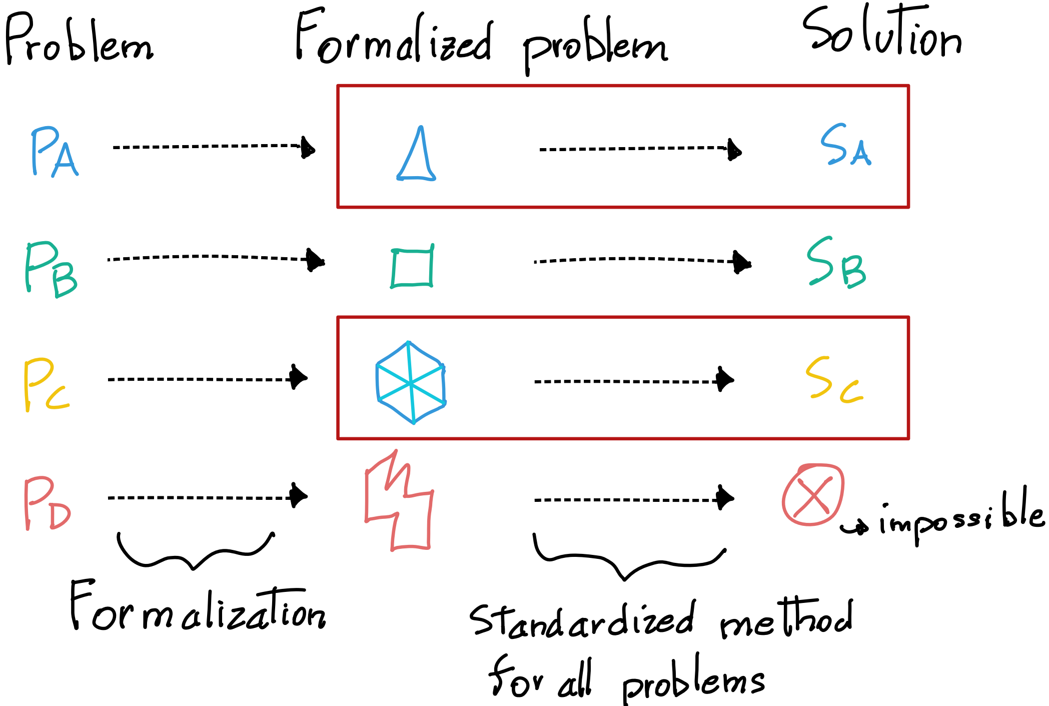
Formalized problem solving



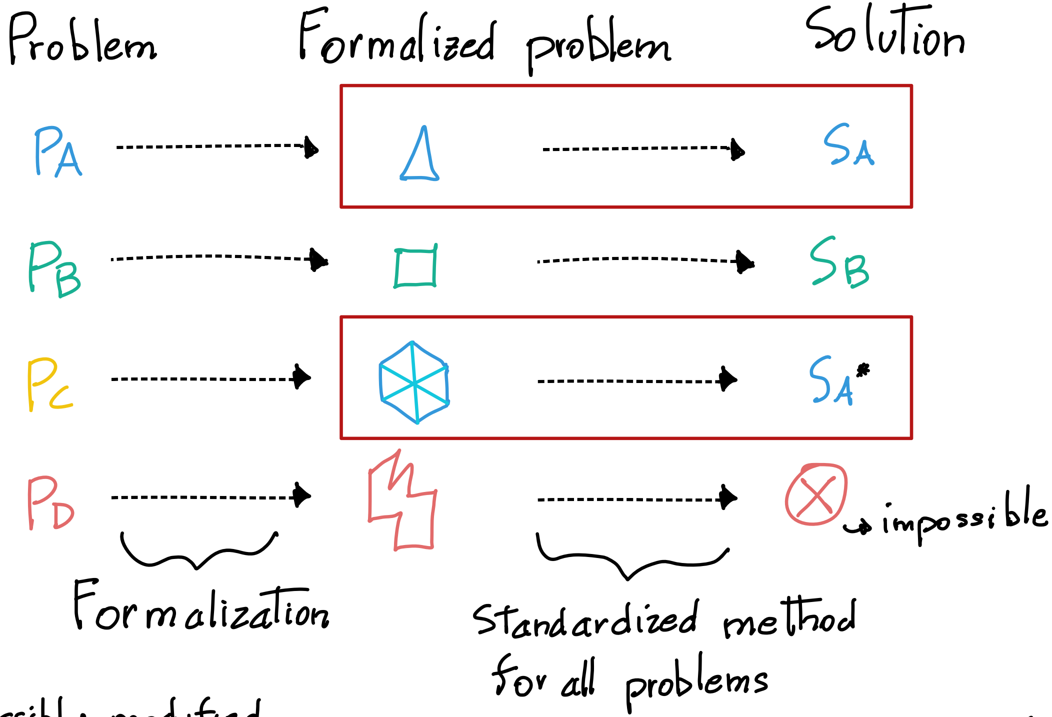
Formalized problem solving



Formalized problem solving

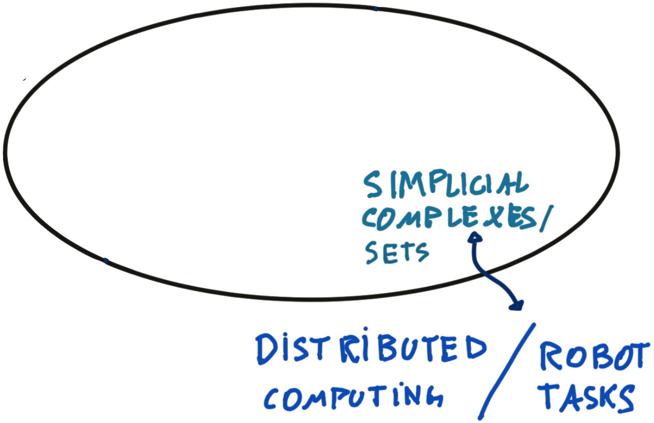


Formalized problem solving

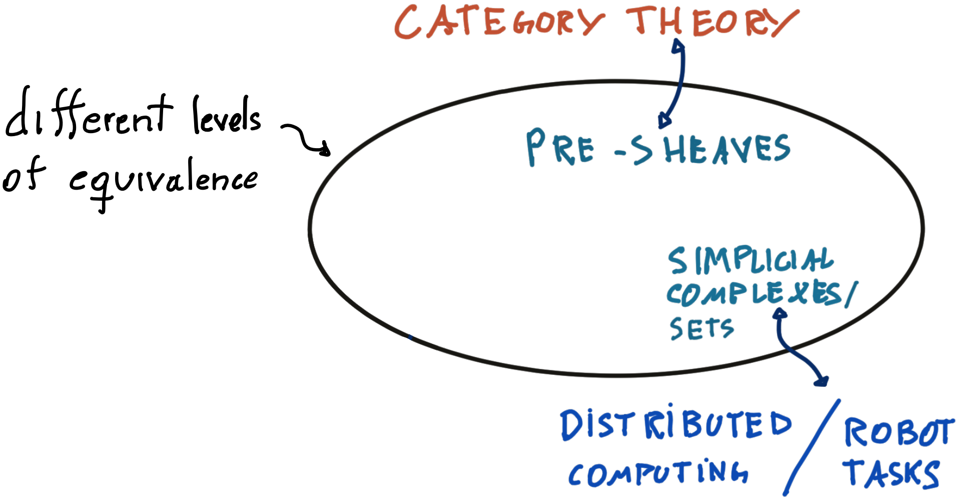


*possibly modified

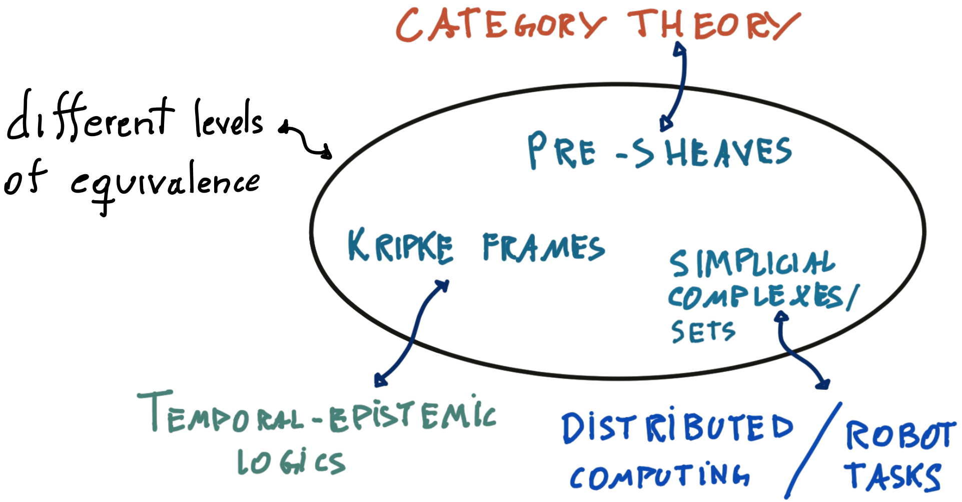
Theoretical background - useful equivalences


SIMPLICIAL COMPLEXES/ SETS
DISTRIBUTED COMPUTING / ROBOT TASKS

Theoretical background - useful equivalences


different levels of equivalence
CATEGORY THEORY
PRE-SHEAVES
SIMPLICIAL COMPLEXES/ SETS
DISTRIBUTED COMPUTING / ROBOT TASKS

Theoretical background - useful equivalences


different levels of equivalence
CATEGORY THEORY
PRE-SHEAVES
KRIPKE FRAMES
SIMPLICIAL COMPLEXES/SETS
TEMPORAL-EPISTEMIC LOGICS
DISTRIBUTED COMPUTING
ROBOT TASKS

Interesting directions

- Common language for robot tasks
- Reusable results from previous works
- Method for conceiving and verifying algorithms
- Multiple points of view and tools for the same problem

Thank you for your attention ☺

PS: Feel free to reach me at hummes@ieee.org

ROBOT TASKS

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\Pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq P(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"']
\end{tikzcd}$$

specification

solvability criteria

protocol complex

ROBOT TASKS
INPUT COMPLEX

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "s"] \arrow[urr, "\delta"'] & &
\end{tikzcd}$$

specification

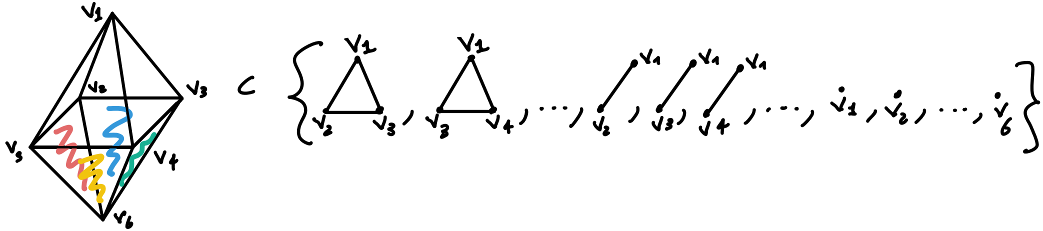
solvability criteria

protocol complex

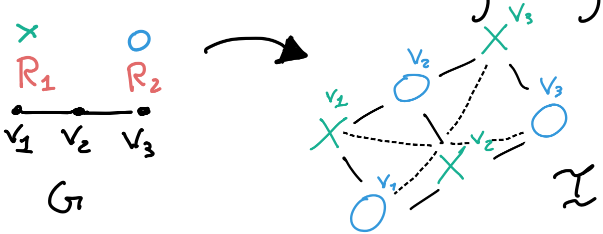
ROBOT TASKS
INPUT COMPLEX

$$\begin{tikzcd} \mathcal{I} & \mathcal{T}\subseteq\mathcal{I}\times\mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\ \mathcal{Q}\subseteq\mathcal{P}(\mathcal{I}) \arrow[u, "s"] \arrow[ur, "\delta"'] & \end{tikzcd}$$

We use simplicial complexes to represent The combination of possible configurations of states



The input complex represents all possible configurations in which The robots can start in a given graph



ROBOT TASKS

TASK DEFINITION

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"'] & &
\end{tikzcd}$$

specification

solvability criteria

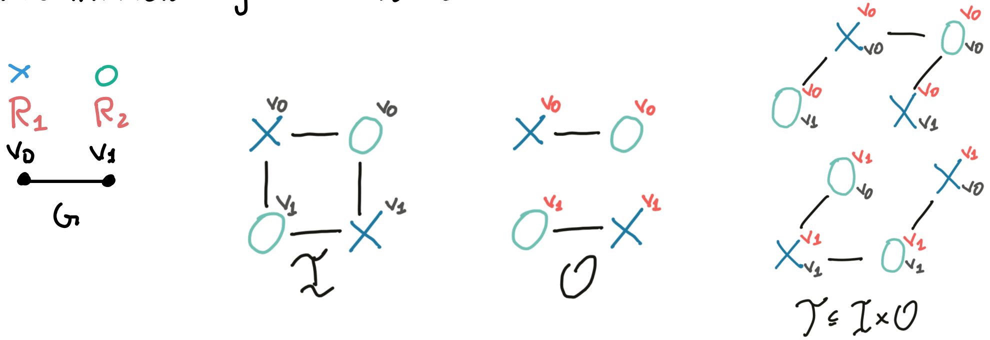
protocol complex

ROBOT TASKS

TASK DEFINITION

$$\begin{tikzcd} \mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\ \mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"'] & \end{tikzcd}$$

A robot task is a triple $(\mathcal{I}, \mathcal{O}, \mathcal{T})$ that defines the pairs of initial and final states respecting the mission objectives in $\mathcal{T}$



Originally defined as the triple $(\mathcal{I}, O, \Delta)$

ROBOT TASKS

ROBOT SPECIFICATION

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq P(\mathcal{I}) \arrow[u, "s"] \arrow[urr, "\delta"'] & &
\end{tikzcd}$$

specification

solvability criteria

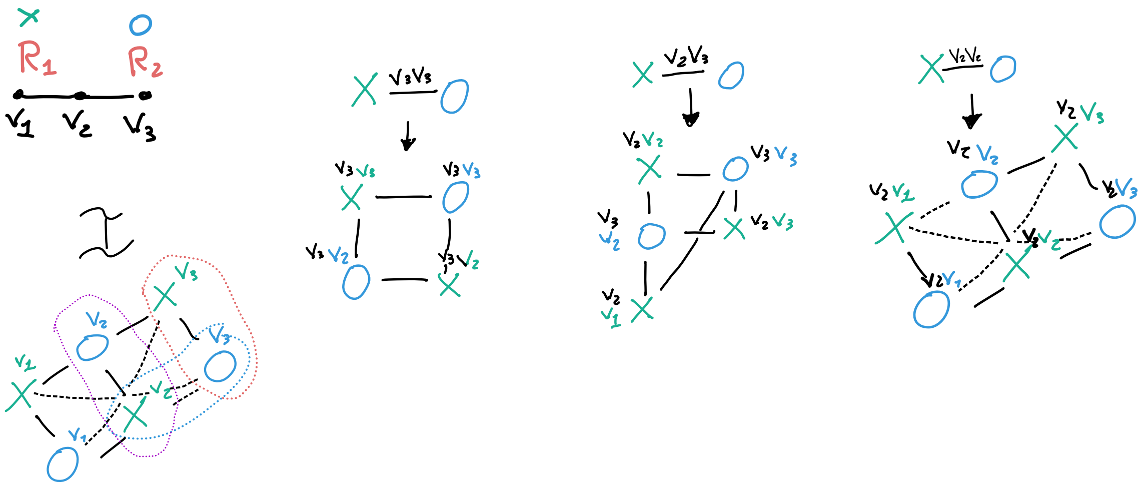
protocol complex

ROBOT TASKS

ROBOT SPECIFICATION

$$\begin{tikzcd}
\mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq P(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"']
\end{tikzcd}$$

The protocol complex encodes all subdivisions of a state according to its possible variations in an execution



ROBOT TASKS

SOLVABILITY

$$\begin{tikzcd}
\mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\Pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"'] &
\end{tikzcd}$$

Task definition

input complex

output complex

specification

solvability criteria

protocol complex

ROBOT TASKS
SOLVABILITY

$$\begin{tikzcd}
\mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"'] &
\end{tikzcd}$$

all possible evolutions of all robots

how all robots may start and end their missions

$$\mathcal{P}(\mathcal{I}) \hookleftarrow \mathcal{Q} \xrightarrow{\delta} \mathcal{T} \xrightarrow{\pi} \mathcal{I} \times \mathcal{O}$$

- has matching initial an final states, added to $\mathcal{Q}$
- respects the robot's sensing limitations, filtered by δ

ROBOT TASKS EXPRESSIVENESS

Task definition

input complex

output complex

$$\begin{tikzcd}
\mathcal{I} & & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[ll, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[urr, "\delta"'] & &
\end{tikzcd}$$

robot specification

solvability criteria

protocol complex

ROBOT TASKS EXPRESSIVENESS

$$\begin{tikzcd}
\mathcal{I} & \mathcal{T} \subseteq \mathcal{I} \times \mathcal{O} \arrow[l, "\pi_{\mathcal{I}}"'] \\
\mathcal{Q} \subseteq \mathcal{P}(\mathcal{I}) \arrow[u, "S"] \arrow[ur, "\delta"']
\end{tikzcd}$$

With this formalization we can express multiple aspects of a robotic mission:

- limitations in the sensors - imposed by the δ map
- crash and freeze failures - naturally expressed in the lower dimensions of the simplicial complexes
- erratic behavior - simple to represent as part of the subdivision rules