Diagnostic Reasoning for Robotics using Action Languages

Esra Erdem[1], Volkan Patoğlu[1], Zeynep G. Saribatur[2]

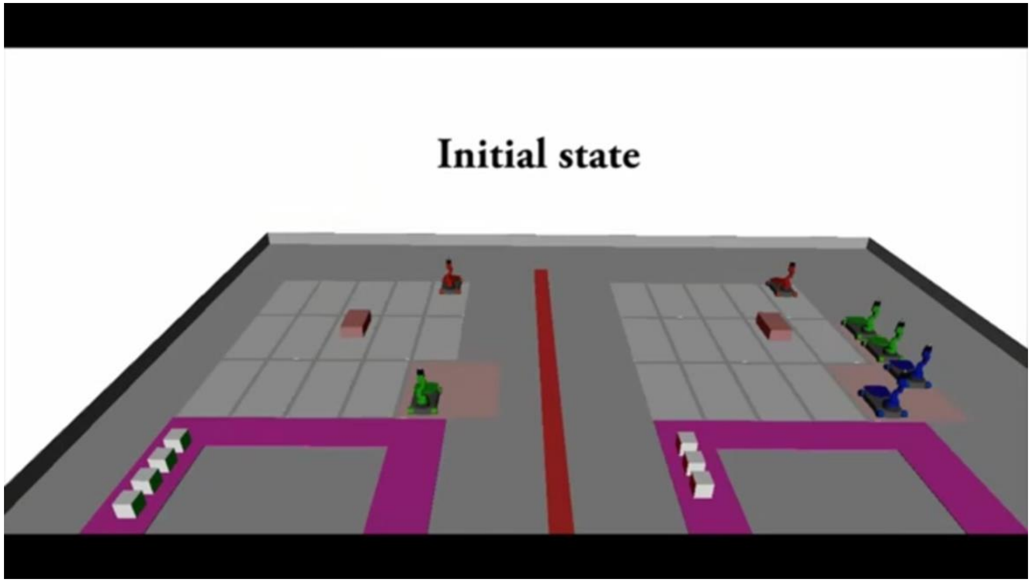
[1] Sabancı University, İstanbul, Turkey
[2] Vienna University of Technology, Vienna, Austria

This research is partially supported by TUBITAK 113M422 and 114E491, 111E116 grants.

CogRobo Cognitive Robotics Laboratory

Sabancı Üniversitesi

Robotic Domains with Multiple Teams of Robots

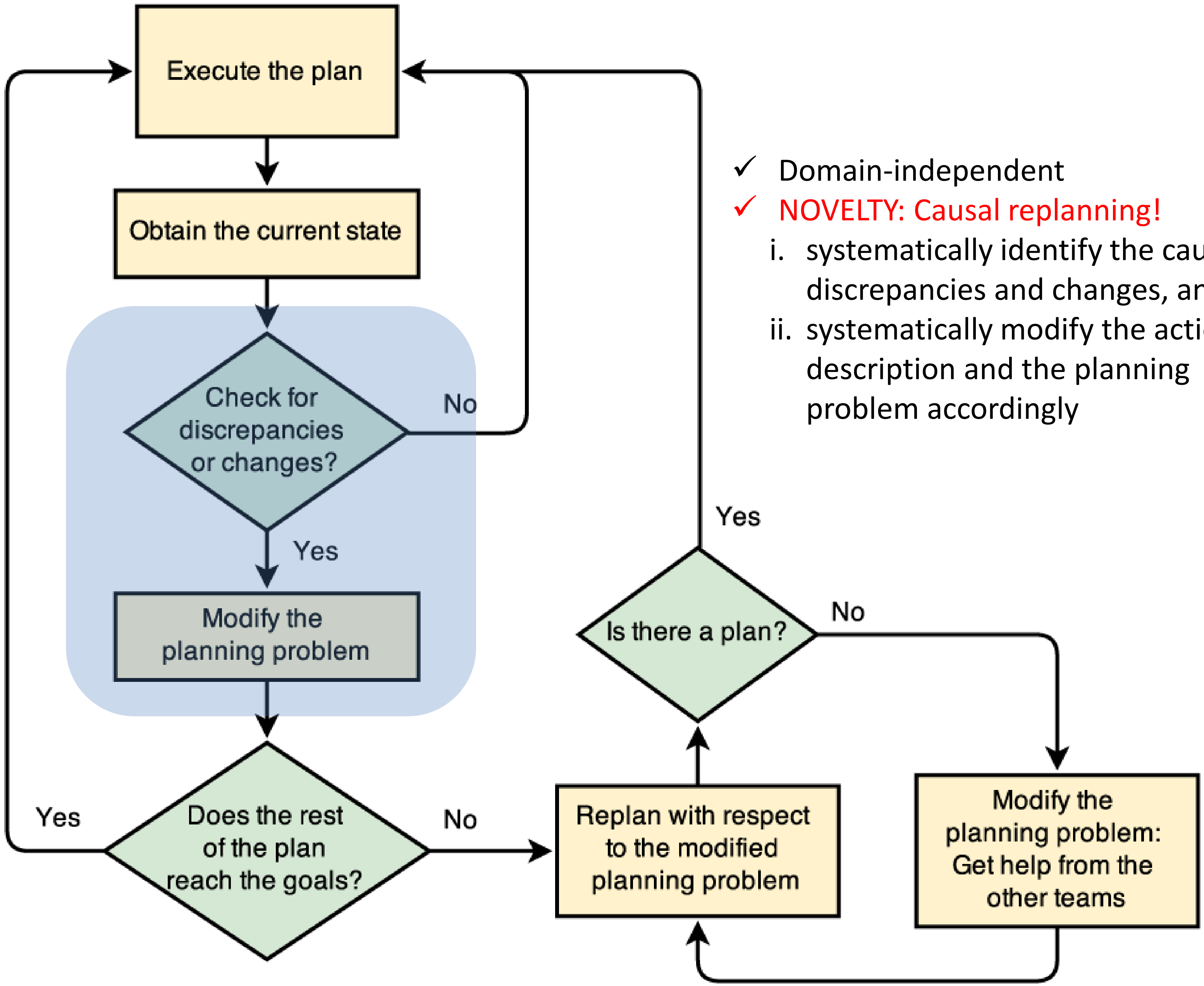


- Teams collaborate with each other for a common goal
- Coordination of teams is needed to use shared resources efficiently

- Each team consists of heterogeneous robots
- Each team has cognitive skills, like hybrid planning with minimum total action cost

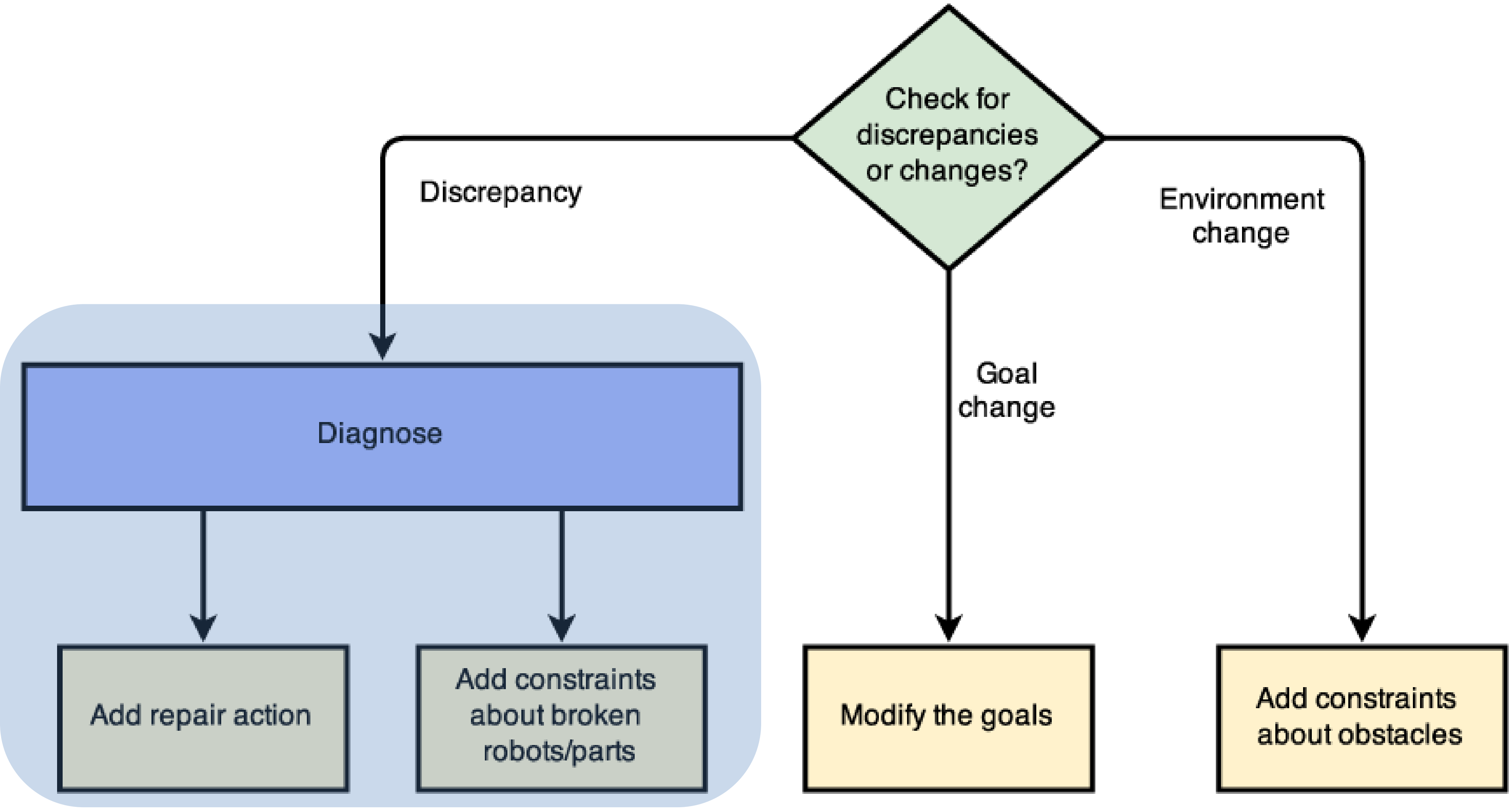
E. Erdem, V. Patoglu, Z. G. Saribatur, P. Schüller, T. Uras, “Finding optimal plans for multiple teams of robots through a mediator: A logic-based approach”, TPLP 13(4-5): 831-846 (2013).
Z. G. Saribatur, E. Erdem, V. Patoglu, “Cognitive factories with multiple teams of heterogeneous robots: Hybrid reasoning for optimal feasible global plans”, Proc. of IROS, 2014.

Plan Execution Monitoring



- ✓ Domain-independent
- ✓ NOVELTY: Causal replanning!
  1. systematically identify the causes of discrepancies and changes, and
  2. systematically modify the action description and the planning problem accordingly

Causal Replanning



NOVELTY: Diagnostic reasoning for replanning!

- Identify causes of discrepancies and modify the planning problem
- Add repair actions to the domain description

Model-Based Diagnosis

Expected behavior of the system

Observations

Hypothesis

$$\underbrace{D \cup O \cup H}$$

Is the logical theory consistent?

Model-Based Diagnosis for Plan Execution

Description of robotic actions and change

Plan execution so far from the initial state

Observations about current state

Hypothesis about broken robots or components

$$\underbrace{D_{diag} \cup S_0^M \cup P_t \cup O_t^M \cup H}$$

Is the logical theory consistent?

If it is consistent then which actions in P_t could not be executed and why not?

Describing Robotic Actions for Diagnostic Reasoning

$$D$$

Description of robotic actions
for robotic planning

Transformation of formulas

- ✓ NOVELTY:
  - no auxiliary actions
  - use of defaults and nondeterminism
  - feasibility checks embedded
- ✓ Systematic and domain independent
- ✓ Polynomial time
- ✓ Correctness proved

$$D_{diag}$$

Description of robotic actions
for diagnosis

Action Languages

- Formalisms for representing and reasoning about actions and change.
- Action description language: C+ [Giunchiglia et al., 2004].

$$\begin{gathered} a \ \textbf{causes} \ F \ \textbf{if} \ G \\ \textbf{nonexecutable} \ A \ \textbf{if} \ G \\ \textbf{caused} \ F \ \textbf{if} \ G \end{gathered}$$

- Action query language: Q [Gelfond & Lifschitz, 1998].

$$\begin{gathered} F \ \textbf{holds at} \ t \\ A \ \textbf{occurs at} \ t \end{gathered}$$

- Automated reasoner: CCalc [McCain 1997].

Transforming an Action Description for Diagnosis

1. For every robot r, add the causal laws

$$\textbf{default } \neg broken(r)$$

$$\textbf{caused } broken(r) \textbf{ if } broken(r) \textbf{ after } \neg broken(r)$$

$$\textbf{caused } broken(r) \textbf{ after } broken(r)$$

Transforming an Action Description for Diagnosis

2. For every (concurrent) action A, add the causal law

$$\textbf{default}\; pre(A)$$

and replace every causal law

$$\textbf{nonexecutable}\; A \;\textbf{if}\; G$$

with the causal law

$$\textbf{caused}\; \neg pre(A) \;\textbf{if}\; G$$

For every primitive action a, replace every causal law

$$a \;\textbf{causes}\; F \;\textbf{if}\; G$$

with the causal law

$$a \;\textbf{causes}\; F \;\textbf{if}\; G, pre(A)$$

Transforming an Action Description for Diagnosis

3. For every primitive action a, for all robots $r_1, \ldots, rm$ that take part in execution of a, replace every causal law

$$a \textbf{ causes } F \textbf{ if } G$$

with the causal law

$$a \textbf{ causes } F \textbf{ if } G, \neg broken(r_1), \ldots, \neg broken(r_m)$$

Proposition 1 Every query satisfied by D is satisfied by D_{diag}.

Diagnosis

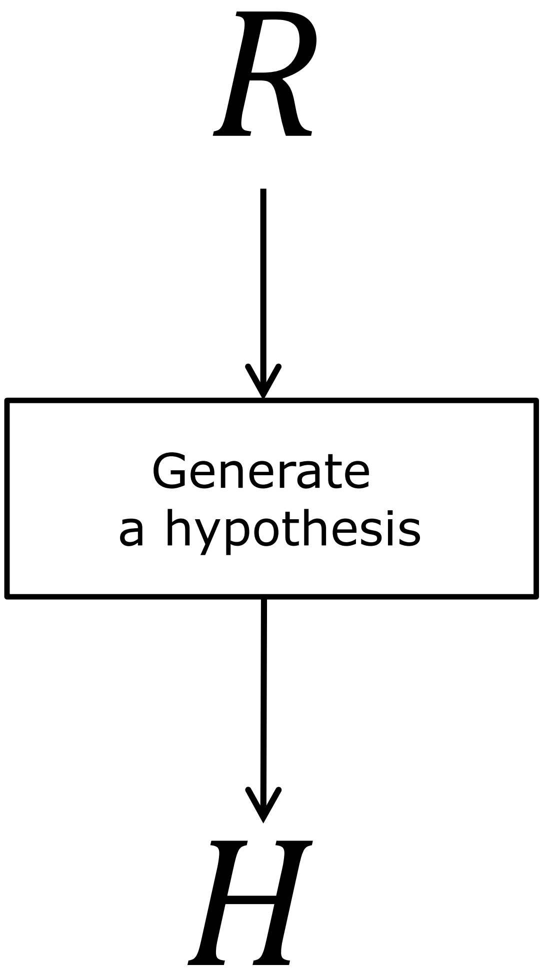
Let's characterize a *diagnosis problem*, *DP*, by $\langle D_{diag}, R, s_0, P_t, o_t \rangle$ where $P_t = \langle A_0,\ldots,A_{t-1} \rangle$.

A *solution* of *DP* is a set $X \subseteq R$ of robots such that D_{diag} satisfies the query

$$s_0 \textbf{ holds at } 0 \wedge \bigwedge_{i=0..t-1} A_i \textbf{ occurs at } i \wedge o_t \textbf{ holds at } t \wedge$$
$$\bigwedge_{r \in X} broken(r) \textbf{ holds at } t \wedge \bigwedge_{r \in R-X} \neg broken(r) \textbf{ holds at } t .$$

We also say that *X* is a *diagnosis* of the discrepancy detected at time *t*.

Generating Hypotheses



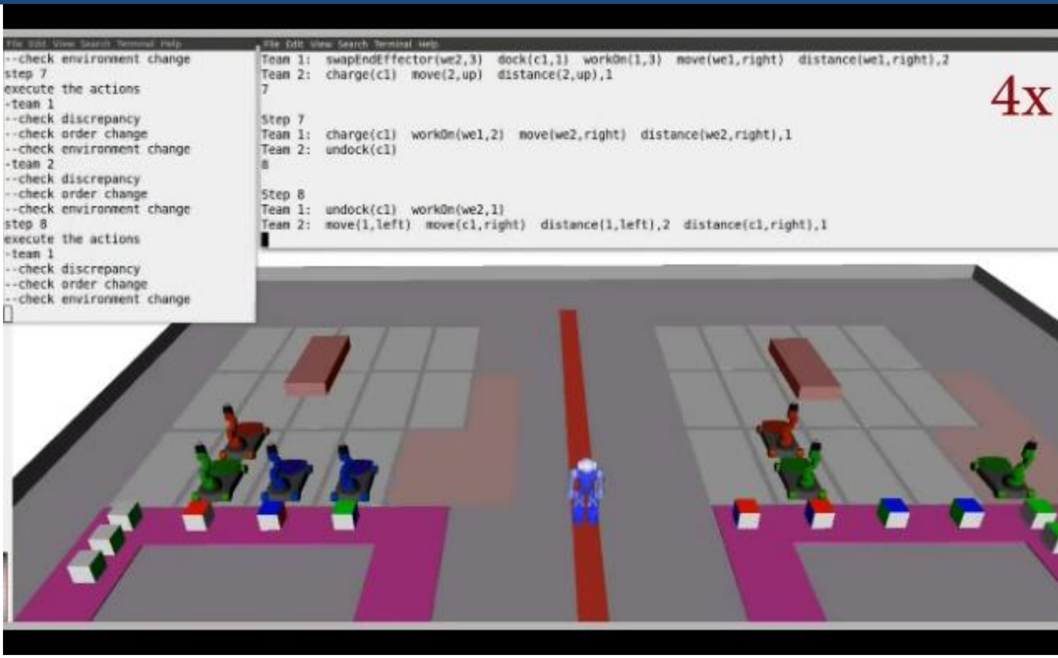
All robots and their components

$$\#minimize\ [1, r : broken(r), r \in R]$$
$$\#maximize\ [w, r: weight(r, w), broken(r), r \in R]$$

- ✓ NOVELTY:
  - Minimality of cardinality is guaranteed, while maximizing the likelihood of hypothesis
  - Learning from experiences with probabilities is utilized in declarative optimization
- ✓ Systematic and domain-independent

Hypothesis about broken robots
and their components

Implementation and Experimentation



Is integrating feasibility checks useful for generating better diagnoses?

YES!

$$\underbrace{D_{diag} \cup S_0^M \cup P_t \cup O_t^M \cup H}$$

Is the logical theory consistent?

False negatives are generated without geometric reasoning.

Is integrating learning useful for generating better diagnoses and faster?

YES!

Scenario	Number of iterations and CPU time [secs]			
	ASP		SAT	
	w/o learning	w/ learning	w/o learning	w/ learning
1 charger, 1 wet, 2 dry Discrepancy at Step 12, Diagnosis cardinality=1	2 6.51 secs	1 3.21 secs	9 52.24 secs	4 26.49 secs
1 charger, 1 wet, 2 dry Discrepancy at Step 8, Diagnosis cardinality=2	2 4.14 secs	1 2.06 secs	21 67.43 secs	11 39.85 secs
1 charger, 1 wet, 2 dry Discrepancy at Step 10, Diagnosis cardinality=3	2 4.28 secs	1 2.14 secs	25 85.88 secs	23 84.16 secs
1 charger, 2 wet, 2 dry Discrepancy at Step 8, Diagnosis cardinality=1	2 7.66 secs	1 3.85 secs	7 83.70 secs	4 42.97 secs
1 charger, 2 wet, 2 dry Discrepancy at Step 12, Diagnosis cardinality=2	2 9.10 secs	1 4.56 secs	24 142.18 secs	12 80.35 secs
1 charger, 2 wet, 2 dry Discrepancy at Step 13, Diagnosis cardinality=3	2 9.86 secs	1 4.93 secs	39 197.01 secs	25 127.25 secs
2 charger, 2 wet, 2 dry Discrepancy at Step 6, Diagnosis cardinality=1	2 6.27 secs	1 3.13 secs	8 97.00 secs	4 49.30 secs
2 charger, 2 wet, 2 dry Discrepancy at Step 14, Diagnosis cardinality=2	2 13.29 secs	1 6.63 secs	31 195.17 secs	13 102.48 secs
2 charger, 2 wet, 2 dry Discrepancy at Step 12, Diagnosis cardinality=3	2 11.57 secs	1 5.78 secs	49 240.90 secs	30 154.85 secs

Does diagnostic reasoning improve replanning?

YES!

Scenario	Number of replanning and total CPU time		
	w/ diagnosis		w/o diagnosis
	w/ repair	w/o repair	
1 charger, 1 wet, 1 dry Discrepancy at Step 5 Diagnosis cardinality=1	3* 10.38	11 9.71	17 82.22
1 charger, 1 wet, 2 dry Discrepancy at Step 12 Diagnosis cardinality=1	3* 0.52	21 18.17	41 5.49
1 charger, 2 wet, 2 dry Discrepancy at Step 8 Diagnosis cardinality=1	2* 4.95	17* 28.51	25 99.68
1 charger, 2 wet, 2 dry Discrepancy at Step 12 Diagnosis cardinality=2	8* 13.41	13 23.04	25 53.14
2 charger, 2 wet, 2 dry Discrepancy at Step 6 Diagnosis cardinality=1	4* 11.78	17 79.38	44 152.07
2 charger, 2 wet, 2 dry Discrepancy at Step 14 Diagnosis cardinality=2	5* 5.00	9* 11.67	36 25.7

* indicates termination with a feasible plan, while others terminate due to non-existence of a feasible plan.

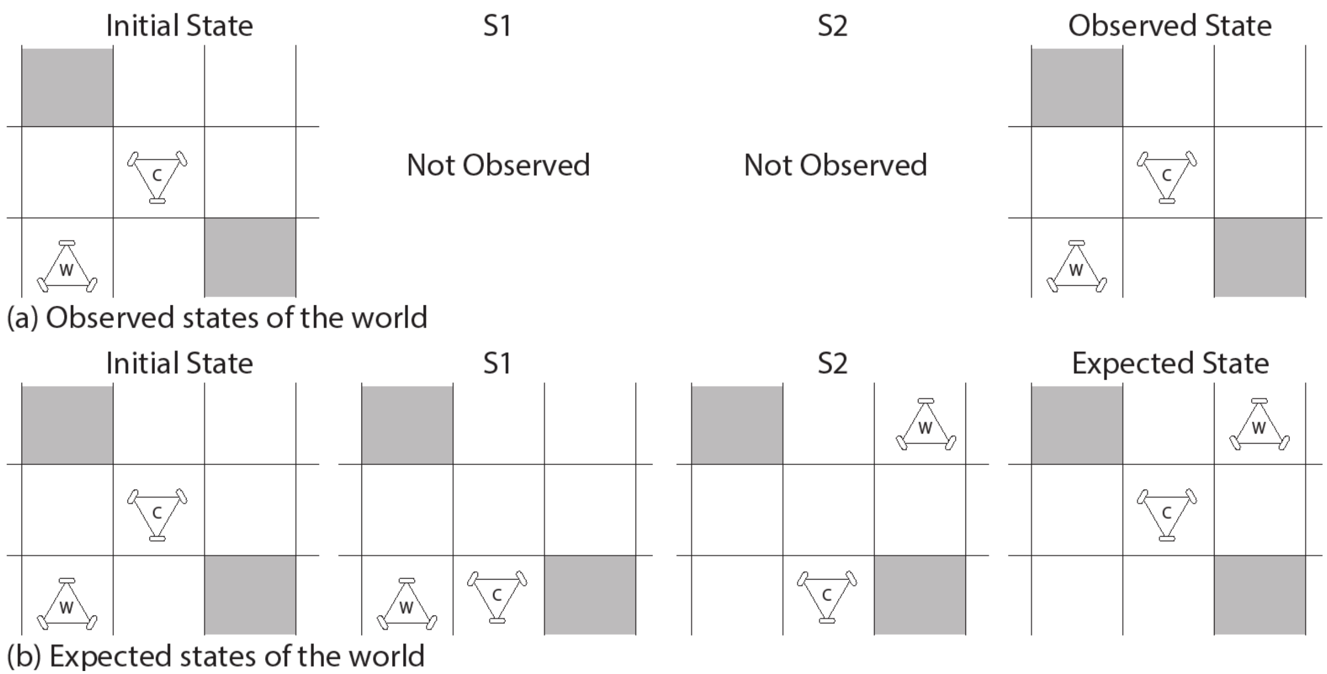
Implementation and Experimentation

Is integrating feasibility checks useful for generating better diagnoses?

YES!

$$\underbrace{D_{diag} \cup S_0^M \cup P_t \cup O_t^M \cup H}_{\text{Is the logical theory consistent?}}$$

False negatives are generated without geometric reasoning.



(a) Observed states of the world

(b) Expected states of the world

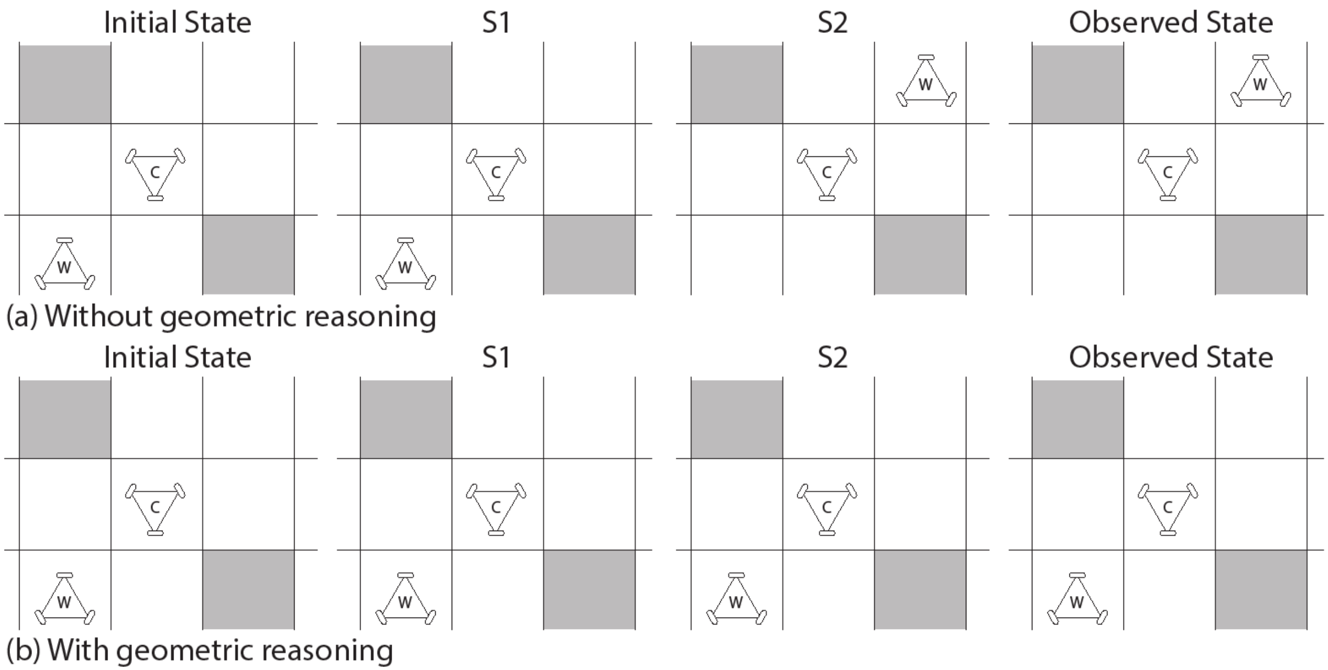
Implementation and Experimentation

Is integrating feasibility checks useful for generating better diagnoses?

YES!

$$\underbrace{D_{diag} \cup S_0^M \cup P_t \cup O_t^M \cup H}_{\text{Is the logical theory consistent?}}$$

False negatives are generated without geometric reasoning.



Implementation and Experimentation

Is integrating learning useful for generating better diagnoses and faster?

YES!

Scenario	Number of iterations and CPU time [secs]			
	ASP		SAT	
	w/o learning	w/ learning	w/o learning	w/ learning
1 charger, 1 wet, 2 dry Discrepancy at Step 12, Diagnosis cardinality=1	2 6.51 secs	1 3.21 secs	9 52.24 secs	4 26.49 secs
1 charger, 1 wet, 2 dry Discrepancy at Step 8, Diagnosis cardinality=2	2 4.14 secs	1 2.06 secs	21 67.43 secs	11 39.85 secs
1 charger, 1 wet, 2 dry Discrepancy at Step 10, Diagnosis cardinality=3	2 4.28 secs	1 2.14 secs	25 85.88 secs	23 84.16 secs
1 charger, 2 wet, 2 dry Discrepancy at Step 8, Diagnosis cardinality=1	2 7.66 secs	1 3.85 secs	7 83.70 secs	4 42.97 secs
1 charger, 2 wet, 2 dry Discrepancy at Step 12, Diagnosis cardinality=2	2 9.10 secs	1 4.56 secs	24 142.18 secs	12 80.35 secs
1 charger, 2 wet, 2 dry Discrepancy at Step 13, Diagnosis cardinality=3	2 9.86 secs	1 4.93 secs	39 197.01 secs	25 127.25 secs
2 charger, 2 wet, 2 dry Discrepancy at Step 6, Diagnosis cardinality=1	2 6.27 secs	1 3.13 secs	8 97.00 secs	4 49.30 secs
2 charger, 2 wet, 2 dry Discrepancy at Step 14, Diagnosis cardinality=2	2 13.29 secs	1 6.63 secs	31 195.17 secs	13 102.48 secs
2 charger, 2 wet, 2 dry Discrepancy at Step 12, Diagnosis cardinality=3	2 11.57 secs	1 5.78 secs	49 240.90 secs	30 154.85 secs

Implementation and Experimentation

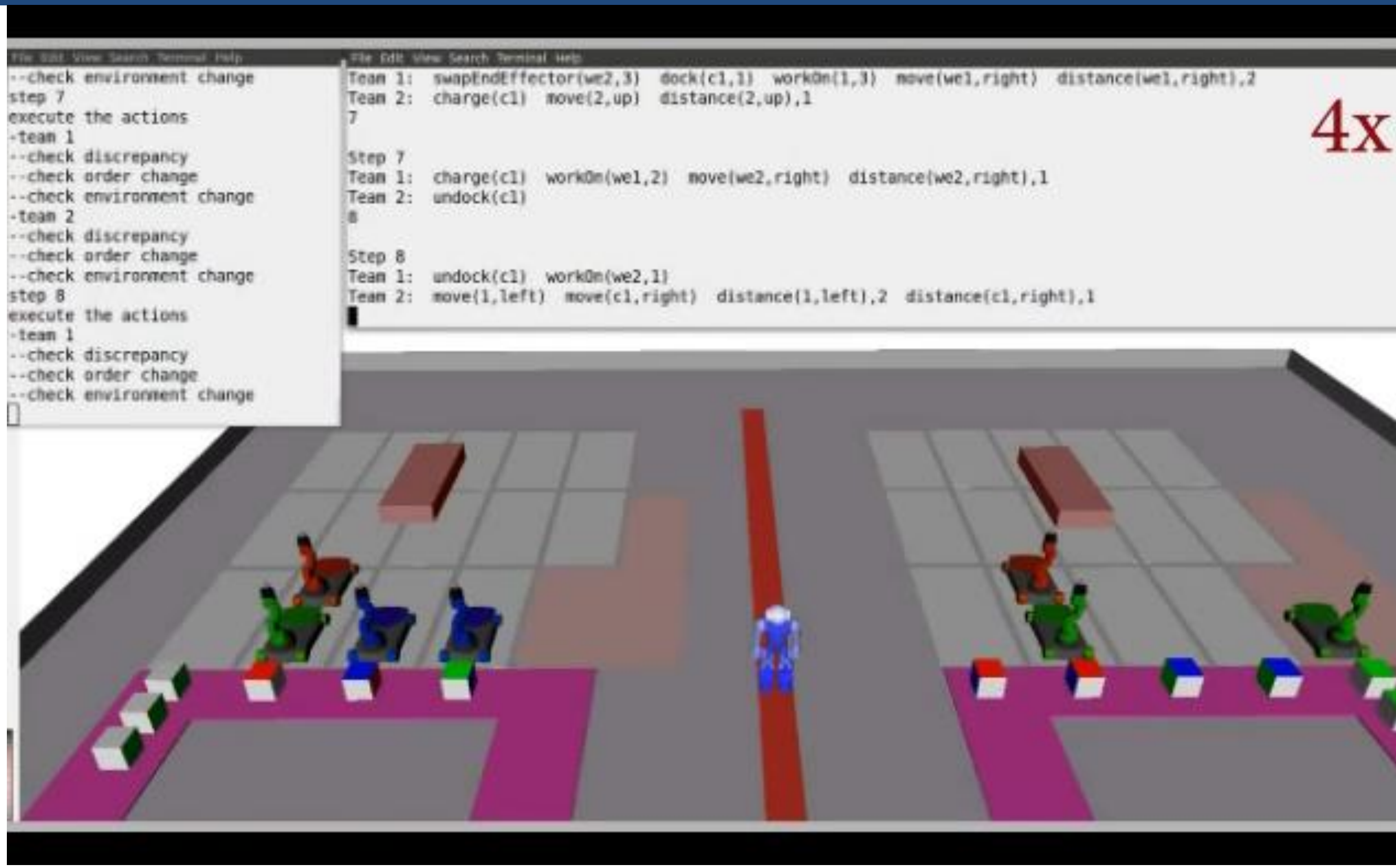
Does diagnostic reasoning improve replanning?

YES!

Scenario	Number of replanning and total CPU time		
	w/ diagnosis		w/o diagnosis
	w/ repair	w/o repair	
1 charger, 1 wet, 1 dry Discrepancy at Step 5 Diagnosis cardinality=1	3* 10.38	11 9.71	17 82.22
1 charger, 1 wet, 2 dry Discrepancy at Step 12 Diagnosis cardinality=1	3* 0.52	21 18.17	41 5.49
1 charger, 2 wet, 2 dry Discrepancy at Step 8 Diagnosis cardinality=1	2* 4.95	17* 28.51	25 99.68
1 charger, 2 wet, 2 dry Discrepancy at Step 12 Diagnosis cardinality=2	8* 13.41	13 23.04	25 53.14
2 charger, 2 wet, 2 dry Discrepancy at Step 6 Diagnosis cardinality=1	4* 11.78	17 79.38	44 152.07
2 charger, 2 wet, 2 dry Discrepancy at Step 14 Diagnosis cardinality=2	5* 5.00	9* 11.67	36 25.7

* indicates termination with a feasible plan, while others terminate due to non-existence of a feasible plan.

Conclusions



Novelties of our diagnostic reasoning framework from the AI and Robotics perspectives:

- It generates diagnoses without introducing auxiliary "break" actions.
- It can optimize these diagnoses.
- It utilizes feasibility checks as needed.
- It utilizes learning from earlier diagnoses and failures.

Erdem, E., Patoglu, V., Saribatur, Z.G.: Integrating hybrid diagnostic reasoning in plan execution monitoring for cognitive factories with multiple robots. Proc. of ICRA (2015).