Integrated Commonsense Reasoning and Interactive Learning in Robotics

Mohan Sridharan
Intelligent Robotics Lab, School of Computer Science, University of Birmingham, UK
Email: m.sridharan@bham.ac.uk

***Abstract*—This paper summarizes work on an architecture for robots that combines the complementary strengths of knowledge-based reasoning and data-driven learning. The architecture supports non-monotonic logical reasoning and probabilistic reasoning with incomplete commonsense domain knowledge. Reasoning triggers and guides learning of previously unknown domain knowledge when needed based on deep learning and reinforcement learning methods. Furthermore, the architecture enables the robot to provide relational descriptions of its decisions and the evolution of beliefs during reasoning and learning. The architecture's capabilities are illustrated and evaluated in simulation and on physical robots.**

I. Motivation

Consider an illustrative *robot assistant* (RA) domain in which a robot has to: (a) deliver target objects to particular people or rooms; and (b) estimate and revise the occlusion of objects and stability of object configurations in a particular room. There is uncertainty in the robot's perception and actuation. The robot's incomplete domain knowledge includes commonsense knowledge, e.g., statements such as "books are usually in the study" that hold in all but a few exceptional circumstances, e.g., cookbooks are in the kitchen. The robot also extracts information from noisy sensor inputs, with quantitative measures of uncertainty, e.g., "I am 90% certain I saw the robotics book in office-1". In addition, the robot has some prior knowledge of object attributes such as $size$, $surface$, and $shape$; some spatial relations between objects; and some axioms governing domain dynamics:

- Placing an object on top of another with an irregular surface results in instability.
- An object can only be in one location at a time.
- An object below another object cannot be picked up.

The robot reasons with the knowledge and observations for inference, planning, and diagnostics. In any practical domain, it will have to revise this knowledge over time; this is often accomplished by data-driven (e.g., deep, reinforcement) learning methods that process observations, labeled datasets, and/or human input. Also, enabling the robot to describe its decisions and the evolution of beliefs at different levels of abstraction will lead to more effective collaboration with humans. Our architecture seeks to support these capabilities by exploiting the complementary strengths of declarative logic programming, probabilistic reasoning, and data-driven interactive learning. We briefly describe the architecture's components below.

II. Architecture Overview

Our baseline architecture for knowledge representation, explainable reasoning and interactive learning, is based on tightly-coupled transition diagrams at different resolutions. It may be viewed as a logician, statistician, and a creative explorer working together; see Figure 1 (left). The different transition diagrams are described using an action language $\mathcal{AL}_d$ [3], which has a sorted signature with statics, fluents, and actions, and supports three types of statements: causal laws, state constraints, and executability conditions; the fluents can be non-Boolean and axioms can be non-deterministic. Depending on the domain and tasks at hand, the robot chooses to plan and execute actions at two specific resolutions, but can construct explanations at other resolutions; we limit our discussion to two resolutions in this paper.

Knowledge representation and reasoning: The coarse resolution domain description comprises system description $\mathcal{D}_c$ of transition diagram τ_c, a collection of $\mathcal{AL}_d$ statements, and history $\mathcal{H}_c$. $\mathcal{D}_c$ comprises sorted signature Σ_c and axioms. For RA domain, Σ_c includes basic sorts such as $place$, $thing$, $robot$, $person$, $object$, cup, $size$, $surface$, and $step$; statics such as $next\_to(place, place)$ and *obj_surface(obj, surface)*; fluents such as $loc(thing, place)$, *obj_rel(relation, object, object)*, and $in\_hand(entity, object)$; and actions such as $move(robot, place)$, *pickup(robot, object)*, *putdown(robot, object, location)*, and $give(robot, object, person)$. Axioms in $\mathcal{D}_c$ include statements such as:

$move(rob_1, P)$ **causes** $loc(rob_1, P)$

$loc(O, P)$ **if** $loc(rob_1, P),\ in\_hand(rob_1, O)$

impossible $give(rob_1, O, P)$ **if** $loc(rob_1, L_1) \neq loc(P, L_2)$

that correspond to a causal law, state constraint, and executability condition respectively.

The history $\mathcal{H}_c$ of a dynamic domain is typically a record of fluents observed to be true or false at a particular time step, and the occurrence of actions at a particular time step. This definition is expanded to represent prioritized defaults describing the values of fluents in the initial state, i.e., statements such as "books are usually in the library; if not there, they are in the office" with the exception "cookbooks are in the kitchen".

To reason with the domain description, we construct program $\Pi(\mathcal{D}_c, \mathcal{H}_c)$ in CR-Prolog, a variant of Answer Set Prolog (ASP) that incorporates consistency restoring (CR) rules [2]. ASP is based on stable model semantics, and supports *default*

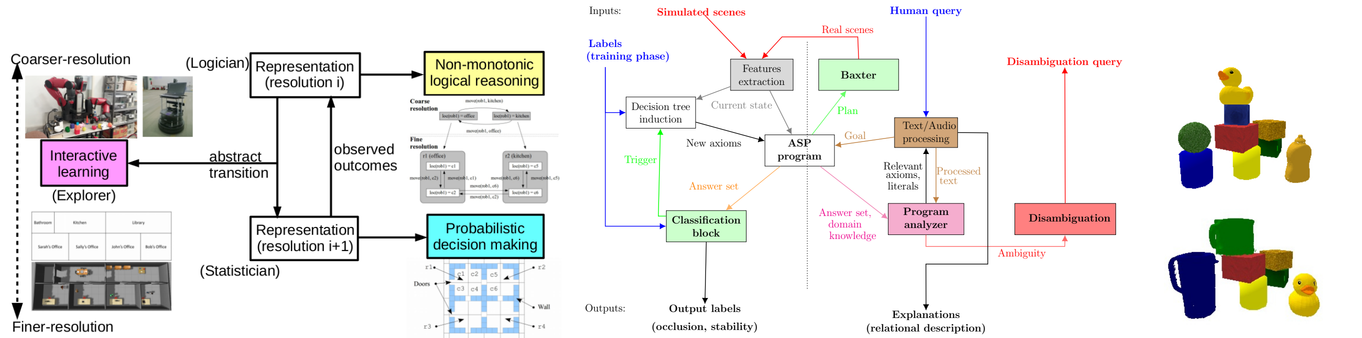

Fig. 1. (left) Architecture combines strengths of declarative programming, probabilistic reasoning, and interactive learning to represent, reason, act, and learn at different resolutions; (center) Non-monotonic logical reasoning triggers and guides deep (and inductive) learning to revise knowledge and provide relational descriptions as explanation; (right) Example images of simulated scene for execution traces.

negation and *epistemic disjunction*, e.g., unlike "$\neg a$" that states *a is believed to be false*, "*not* a" only implies *a is not believed to be true*, i.e., each literal can be true, false or "unknown". ASP represents constructs difficult to express in classical logic formalisms and supports non-monotonic logical reasoning. An *answer set* of Π represents the beliefs of the robot associated with Π. Tasks such as computing entailment, planning, and diagnostics can be reduced to computing answer sets of Π; we do so using the SPARC system [1].

For any given goal, reasoning at the coarse-resolution provides a plan of *abstract actions*. To implement the abstract actions, we construct a fine-resolution system description $\mathcal{D}_f$ defined as a *refinement* of $\mathcal{D}_c$. This definition ensures that for any given abstract transition between two states $\in \tau_c$, there is a path in τ_f between a refinement of the two states. In the RA domain, the robot would (for example) reason about grid cells in rooms and parts of objects, attributes that were previously abstracted away by the designer. Since the robot interacts with the physical world at the finer resolution, we introduce a *theory of observation* in $\mathcal{D}_f$, specifically *knowledge-producing* actions and fluents to sense the value of domain fluents. Next, $\mathcal{D}_f$ is *randomized* to model non-determinism ($\mathcal{D}_{fr}$). Since reasoning with $\mathcal{D}_{fr}$ becomes computationally unfeasible for complex domains, we enable the robot to automatically *zoom* to $\mathcal{D}_{fr}(T)$, the part of $\mathcal{D}_{fr}$ *relevant* to any given abstract transition T. Reasoning with $\mathcal{D}_{fr}(T)$ provides a sequence of concrete actions that implement T. This reasoning also incorporates relevant probabilistic models of the uncertainty in perception and actuation, e.g., we have used hierarchical probabilistic sequential decision making algorithms. Fine-resolution outcomes with a high probability are committed as statements known with complete certainty. Reasoning with these outcomes provides coarse-resolution outcomes that are added to $\mathcal{H}_c$ for further reasoning. Please see [11] for details.

Interactive learning: Reasoning with incomplete domain knowledge to achieve desired goals (e.g., fetch target objects) or perform desired estimation tasks (e.g., classifying occlusion of objects or stability of object structures) can result in incorrect/suboptimal outcomes. State of the art methods for learning previously unknown actions and axioms, or object models for estimation tasks, are based on deep networks. They often require many labeled examples; it is difficult to provide such examples in complex domains or to interpret the decisions of such "end to end" data-driven methods.

Figure 1(center) is an overview of the interactive learning and explainable reasoning components. The main sensor inputs for these components (and the architecture) are RGB/D images. These images are processed to extract spatial relations (based on learned grounding of prepositions [5]) and other attributes that are encoded as ASP statements. The robot first uses ASP-based logical reasoning to complete the desired (e.g., planning, estimation) tasks. If this reasoning does not provide any outcome (e.g., no plan to reach goal), or provides an incorrect outcome (e.g., incorrect classification label on training image), this is considered to indicate that the knowledge is incomplete or incorrect, which triggers learning.

The architecture has two schemes for learning and knowledge revision. The first scheme is used in the context of planning; relational reinforcement learning and decision-tree induction are used to learn actions and axioms from human descriptions of desired behavior, or observations obtained through active exploration or reactive action execution in response to the unexpected outcomes. Reasoning automatically limits this learning to states, actions, and observations relevant to the task(s) and goal(s) at hand; see [9] for details. The second scheme is used in the context of estimation tasks on input images. Reasoning with domain knowledge helps the robot automatically identify relevant regions of interest (ROIs) from the corresponding images, using information from these ROIs to efficiently train a deep neural network for the estimation tasks. This information is also used to incrementally learn decision trees summarizing the robot's experiences, with axioms induced from branches of these trees being merged with existing axioms for reasoning; see [8] for details.

Explainable reasoning: We consider an "explanation" to be a relational description of the robot's decisions or beliefs in terms of the domain attributes and robot actions. The explainable reasoning component of our architecture is based on a *theory of explanations* that comprises (i) claims about representing, reasoning with, and learning knowledge to support explanations; (ii) a characterization of explanations along three axes based on abstraction, specificity, and verbosity; and (iii) a methodology for constructing explanations [10].

The robot processes human verbal or textual input using existing natural language processing tools and an underlying controlled vocabulary to identify the type of query (e.g., descriptive, contrastive, counterfactual). The architecture enables the robot to automatically trace the evolution of desired beliefs (and the application of relevant axioms) to identify literals relevant to the query; these are used to construct the answer presented to the human user. The human can interactively obtain the answer at the desired level of abstraction [10], and the robot can construct and pose clarification questions to resolve ambiguities in the human query [7].

III. Execution Traces

Consider the following execution traces of our architecture.

Execution Example 1. *[Planning and Learning]*
The robot in the RA domain is in the $study$; it is asked to bring a cup to the $study$, i.e., the goal state contains: $loc(C, study),\ not\ in\_hand(rob_1, C)$, where C is a cup.

- The computed plan of abstract actions is:
  $$move(rob_1, kitchen),\quad pickup(rob_1, C),$$
  $$move(rob_1, study),\quad putdown(rob_1, C)$$
  where C is a cup. This plan uses the default knowledge that cups are usually in the $kitchen$ that is next to $study$.
- To implement each abstract transition T, the robot zooms to $\mathcal{D}_{fr}(T)$. For instance, only cells in the $study$ and the $kitchen$ are relevant to the first $move$ action; irrelevant objects and locations are ignored.
- The zoomed description is used to obtain a probabilistic policy that is invoked repeatedly to execute a sequence of concrete actions that implements the abstract action, e.g., robot is in a cell in the $kitchen$ after first $move$. Other actions are executed in a similar manner.
- Suppose that the robot's attempt to pick up a cup in the kitchen failed. The robot observes that the cup is $heavy$ while its arm is $light$. The robot can then learn the executability condition:
  $$\textbf{impossible}\ pickup(rob_1, C)\ \textbf{if}\ arm(rob_1, light),\ obj\_weight(C, heavy)$$
  It is also possible to learn and merge other kinds of axioms with the existing knowledge.

Next, consider the robot's ability to provide explanations.

Execution Example 2. *[Explanation and Disambiguation]*
Consider a robot that has executed the plan from the previous example. When asked to provide a more detailed description, the robot revises the abstraction level, e.g., uses the fine-resolution description.

- **Human:** "Please describe the executed plan in detail."
  Robot: "I moved to cell c_2 in the $kitchen$. I picked the large cup by its handle from the counter [...] I moved to cell c_4 of the $study$. I put the cup down on the red table."

Consider the simulated scenario in Figure 1(right, bottom).

- **Human**: "Move the yellow object on the green cube."
  There is ambiguity in the reference to an yellow object. Since the yellow cube is already on the green cube, and the yellow cylinder is below other objects, the robot poses the clarification question.
  Robot: "Should I move yellow duck on the green cube?"
  Human: "No. Move yellow cylinder on the green cube."
- The robot computes the plan: *pick up the green mug; put the green mug on the table; pick up the red cube; put the red cube on the table; pick up the yellow cube; put the yellow cube on the table; pick up the yellow cylinder; put the yellow cylinder on the green cube*.
  Human: "Why do you want to pick up the green mug?"
  Robot: "I have to place the yellow cylinder on the green cube, and the yellow cylinder is below the green mug."
- The robot can also trace beliefs and axioms to answer questions after plan execution.
  Human: "Why did you not pick up red cube at step1?"
  Robot: "Because the red cube is below the green mug."
  Human: "Why did you move yellow cube to the table?"
  Robot: "I had to put the yellow cylinder on top of the green cube. The green cube was below the yellow cube."

Experimental results in simulation and on physical robots, are described in [4, 6, 8, 9, 10, 11].

IV. Discussion

The examples illustrate some advantages of our architecture.

- First, once the designer has provided the domain-specific information (e.g., for refinement), planning, diagnostics, and plan execution can be automated. The formal coupling between the resolutions helps introduce more complex theories in the coarse-resolution, and to exploit the complementary strengths of non-monotonic logical reasoning and probabilistic reasoning.
- Second, exploiting the interplay between knowledge-based reasoning and data-driven learning helps focus attention automatically to the relevant knowledge at the appropriate resolution, thus improving the reliability and efficiency of reasoning and learning.
- Third, it is easier to understand and modify the observed behavior than with architectures that consider all the available knowledge or only support probabilistic reasoning. The robot is able to provide relational descriptions of its decisions and the evolution of beliefs, automatically resolving any ambiguities in the human query.
- Fourth, there is smooth transfer of control and relevant knowledge between components of the architecture, and confidence in the correctness of the robot's behavior. Also, the underlying methodology can be used with different robots and domains.

Acknowledgements

This work is the result of research threads pursued in collaboration with Ben Meadows, Tiago Mota, Heather Riley, Rocio Gomez, Michael Gelfond, Shiqi Zhang, and Jeremy Wyatt. This work was supported in part by the U.S. ONR Awards N00014-13-1-0766, N00014-17-1-2434 and N00014-20-1-2390, AOARD award FA2386-16-1-4071, and U.K. EPSRC award EP/S032487/1.

References

[1] Evgenii Balai, Michael Gelfond, and Yuanlin Zhang. Towards Answer Set Programming with Sorts. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, Corunna, Spain, September 15-19, 2013.

[2] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice, Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan Claypool Publishers, 2012.

[3] Michael Gelfond and Daniela Inclezan. Some Properties of System Descriptions of AL_d. *Journal of Applied Non-Classical Logics, Special Issue on Equilibrium Logic and Answer Set Programming*, 23(1-2):105–120, 2013.

[4] Rocio Gomez, Mohan Sridharan, and Heather Riley. What do you really want to do? Towards a Theory of Intentions for Human-Robot Collaboration. *Annals of Mathematics and Artificial Intelligence, special issue on commonsense reasoning*, 89:179–208, Februray 2021.

[5] Tiago Mota and Mohan Sridharan. Incrementally Grounding Expressions for Spatial Relations between Objects. In *International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, July 2018.

[6] Tiago Mota and Mohan Sridharan. Commonsense Reasoning and Knowledge Acquisition to Guide Deep Learning on Robots. In *Robotics Science and Systems*, Freiburg, Germany, June 22-26, 2019.

[7] Tiago Mota and Mohan Sridharan. Answer me this: Constructing Disambiguation Queries for Explanation Generation in Robotics. In *IEEE International Conference on Development and Learning (ICDL)*, August 23-26, 2021.

[8] Tiago Mota, Mohan Sridharan, and Ales Leonardis. Integrated Commonsense Reasoning and Deep Learning for Transparent Decision Making in Robotics. *Springer Nature Computer Science*, 2(242):1–18, 2021.

[9] Mohan Sridharan and Ben Meadows. Knowledge Representation and Interactive Learning of Domain Knowledge for Human-Robot Collaboration. *Advances in Cognitive Systems*, 7:77–96, December 2018.

[10] Mohan Sridharan and Benjamin Meadows. Towards a Theory of Explanations for Human-Robot Collaboration. *Kunstliche Intelligenz*, 33(4):331–342, December 2019.

[11] Mohan Sridharan, Michael Gelfond, Shiqi Zhang, and Jeremy Wyatt. REBA: A Refinement-Based Architecture for Knowledge Representation and Reasoning in Robotics. *Journal of Artificial Intelligence Research*, 65: 87–180, May 2019.