From an Agent Logic to an Agent Programming Language for Partially Observable Stochastic Domains

Gavin Rens
CSIR Meraka Institute
and
University of KwaZulu-Natal, South Africa

Abstract

Broadly speaking, my research concerns combining logic of action and POMDP theory in a coherent, theoretically sound language for agent programming. We have already developed a logic for specifying partially observable stochastic domains. A logic for reasoning with the models specified must still be developed. An agent programming language will then be developed and used to design controllers for robots.

1 Motivation for Research

Robots and intelligent agents have to cope with complex and noisy environments. Part of the problem is that agents' actuators and sensors are noisy, causing uncertainty in their action and perception.

To reason and make sensible decisions, they must first have a model of their environment, even though their knowledge is incomplete and hazy. And as the agent or robot executes actions and makes (noisy) observations, it must update its beliefs (subjective knowledge base) accordingly. An agent designer cannot write a plan or recipe for every one of its goals and every possible situation the agent may be in—at least, not for realistic domains. An agent should have a means of *generating* plans for any situation and goal [Thrun *et al.*, 2005].

Using a formal logical language for (commonsense) reasoning about acting and sensing is widely accepted [Levesque and Pirri, 1999; Mueller, 2006]. But classical logics do not deal well with fine-grained uncertainty. POMDP theory [Monahan, 1982] has proven to be a good general framework for formalizing *dynamic, stochastic* systems, with the required fine granularity. Moreover, POMDPs add a notion of preference for informing agent behavior. A drawback of traditional POMDP models though, is that they do not and cannot include information about general facts and laws. Moreover, *axioms* describing the dynamics of a domain cannot be written in POMDP theory. This is what logics are good at [Levesque and Pirri, 1999].

A logic-based language such as Golog [Levesque *et al.*, 1997] has proven effective for robot programming, especially because of its ability to constrain the action search space by axiomatic/logical statements. DTGolog [Boutilier *et al.*, 2000] has the benefits of Golog, and furthermore can deal with domains modeled as Markov decision processes (MDPs). PODTGolog [Rens, 2010] is a Golog dialect attempting to deal with partially observable MDP (POMDP) environments. PODTGolog has not been given a mathematical foundation, though.

Therefore, given the benefits of a logical language for knowledge representation and for planning (as theorem proving), and given how well POMDP theory is suited to modeling partially observable stochastic domains, it seems like a good idea to combine the two formalisms for knowledge-based robots in noisy and uncertain environments.

My Ph.D research programme is a study in cognitive robotics, particularly to develop the theory of a formal logic for cognitive robots living in noisy, nondeterministic domains, and to move the theory into practice by implementing a robot controller with a programming language based on the logic. The research programme will be concluded by performing experiments on a robot programmed with the agent programming language, and analyzing its performance.

2 Work Done

Modal logic [Hughes and Cresswell, 1996] is considered to be well suited to reasoning about beliefs and changing situations. To facilitate the correspondence between POMDPs and modal logic, we required *observation objects* in the logic to correspond to a POMDP's set of observations. For this purpose, as the first work for my doctoral research, the Logic of Actions and Observations (LAO) [Rens *et al.*, 2010], a multimodal logic, was developed: it has explicit observations as first-class elements. Modal logics can typically not express fine-grained degrees of uncertainty and belief. The second step in my research was thus to develop a logic called SLAOP. SLAOP can specify an agent domain, including, executability of actions, perceivability of observations, action transition probabilities, and rewards for and costs of actions as motivations to direct an agent's behavior.

To get a flavor of the language of SLAOP, I briefly present its syntax: The vocabulary contains four sorts: a finite set of *fluents* (alias *propositional atoms*) $\mathfrak{P} = \{p_1, \ldots, p_n\}$, a finite set of names of atomic *actions* $\mathfrak{A} = \{\alpha_1, \ldots, \alpha_n\}$, a finite set of names of atomic *observations* $\Omega = \{\varsigma_1, \ldots, \varsigma_n\}$, and a countable set of names $\mathfrak{Q} = \{q_1, q_2, \ldots\}$ of *rational numbers* in $\mathbb{Q}$. Let $\alpha, \alpha' \in \mathfrak{A}$, $\varsigma, \varsigma' \in \Omega$, $q \in (\mathfrak{Q} \cap (0, 1])$, $r, c \in \mathfrak{Q}$ and $p \in \mathfrak{P}$. The language of SLAOP, is the least set of Φ defined by the grammars:

$$\begin{aligned}\varphi &::= p \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi.\\ \Phi &::= \varphi \mid \neg\Phi \mid \Phi \wedge \Phi \mid [\alpha]_q\varphi \mid (\varsigma \mid \alpha)_q \mid (\varsigma \mid \alpha)^{\diamond} \mid\\ &\qquad \alpha = \alpha' \mid \varsigma = \varsigma' \mid Reward(r) \mid Cost(\alpha, c).\end{aligned}$$

$[\alpha]_q\varphi$ is read 'The probability of reaching a world in which φ holds after executing α, is equal to q'. $(\varsigma \mid \alpha)_q$ can be read 'The probability of perceiving ς is equal to q, given α was performed'. $(\varsigma \mid \alpha)^{\diamond}$ is read 'It is possible to perceive ς', given α was performed'. $Reward(r)$ can be read 'The reward for being in the current situation is r units and we read $Cost(\alpha, c)$ as 'The cost for executing α is c units'.

My supervisors and I have developed semantics for LAO and SLAOP, based on multi-modal logic, and we have developed corresponding tableau calculi to determine validity of sentences in LAO and SLAOP.

It is well known that solving POMDPs optimally is intractable. To address this, we have proposed a method for finding approximate solutions which is potentially tractable, given reasonable assumptions [Rens, 2011].

3 Work Still to be Done

SLAOP is a *specification* logic; it is not a logic for decision-making proper. In forthcoming work we intend developing the Logic and Utility based Agent Planning (LUAP) language, which will employ domain specifications written in SLAOP for 'meta' reasoning. LUAP will be able to reason over sentences involving arbitrary sequences of actions and associated observations. LUAP will maintain/update a belief state (i.e., a probability distribution over possible worlds) according to POMDP theory. Agents designed with LUAP will thus be able to make decisions according to their latest (subjective) beliefs.

LUAP will also have a $Policy(\alpha, h, \Pi, R)$ predicate which holds in case the policy Π generated from program α has horizon h and utility R . Program α is actually a compound action; sequences of atomic actions, while loops, etc. Policies will be sought on the basis of expected utility theory, maximizing expected utility according to the rewards and costs specified in the language of SLAOP. Inspiration for LUAP—especially the definition of $Policy(\cdot)$—has come from DT-Golog [Boutilier *et al.*, 2000] and PODTGolog [Rens, 2010].

Given a formalization $\mathcal{K}$ of a scenario in which a robot wants to grab a can of oil and drink the contents, the robot may have the following queries:

- Is there a reasonable probability (greater than 0.25) that the oil-can is heavy? That is, does $\{weigh, obsHeavy\}\mathbf{B}(\top) > 0.25$ follow from $\mathcal{K}$?
- Is the utility of grabbing the can, then drinking the contents, at least 11 reward units? That is, does $\mathbf{E}\{grab\}\,\{drink\} \geq 11$ follow from $\mathcal{K}$?

Finally, in this research programme, a robot cognition design (CogniD) language will be proposed. CogniD will be based on the formal semantics of LUAP, but it will be oriented towards implementation, that is, for robot programming, and for design of robot controllers. The CogniD interpreter will most likely include a simple sense-plan-act architecture and it will have some optimizations to make the robot behave effectively with respect to certain benchmark tasks.

4 Summary

The research program can be seen as a study of a robotic cognitive system—from the development of the foundation logic to the development of a programming language and its implementation as a robot controller. As such, the research will have to neglect many avenues of interest and importance. For instance, topics in cognitive robotics such as concurrent action, error recovery, goal reconsideration, commonsense reasoning and learning will not be tackled. It is my aim, though, that the CogniD programming language be easy to use and efficient enough for researchers to want to use it in practice. I also hope that my work forms the foundation of another perspective for further research in cognitive robotics.

References

[Boutilier *et al.*, 2000] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proc. of AAAI-00 and of IAAI-00*, pages 355–362. AAAI Press, Menlo Park, CA, 2000.

[Hughes and Cresswell, 1996] G. Hughes and M. Cresswell. *A New Introduction to Modal Logic*. Routledge, New York, NY, 1996.

[Levesque and Pirri, 1999] H. Levesque and F. Pirri. *Logical Foundations for Cognitive Agents: Contributions in Honor of Ray Reiter*. Springer Verlag, 1999.

[Levesque *et al.*, 1997] H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl. GOLOG: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31(1–3):59–84, 1997.

[Monahan, 1982] G. E. Monahan. A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1–16, 1982.

[Mueller, 2006] E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann Publishers, 2006.

[Rens *et al.*, 2010] G. Rens, I. Varzinczak, T. Meyer, and A. Ferrein. A logic for reasoning about actions and explicit observations. In Jiuyong Li, editor, *Proceedings of the 23rd Australasian Joint Conference*, volume 6464 of *Lecture Notes in Artificial Intelligence*, pages 395–404, 2010.

[Rens, 2010] G. Rens. A belief-desire-intention architecture with a logic-based planner for agents in stochastic domains. Master's thesis, School of Computing, University of South Africa, 2010.

[Rens, 2011] G. Rens. Towards real-time policy generation with belief state clustering for dynamic POMDP environments. Technical Report KRR-11-02, Knowledge Representation and Reasoning, CSIR Meraka, Pretoria, South Africa, March 2011.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Massachusetts/England, 2005.