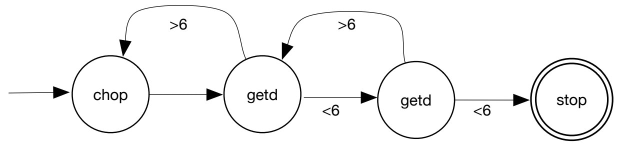

Figure 3: a controller for chopping the tree with noisy sensing

A plan that is epistemically correct for this goal is given in Figure 2. We only argue for the initial world S_0. It also depicts a possible execution path of the plan using V^*. Let us suppose $d(S_0) = 1$. The first action advised by the controller is *chop*, and suppose the action instantiates as $chop(1,0)$. Then $d(do(chop(1,0), S_0)) = 1$; so its still standing, and also, the agent accords a belief of .9 to $[0,9]$ that corresponds to a successful move, and a belief of .1 to its failure. The noise-free sensor naturally returns *up*. The controller advises *chop* again, and suppose this time, the action instantiates as $chop(1,1)$.

Incidentally, even without a sensor reading, the degree of belief in $d < 10$ is $> .9$ because the only branch that still entertains d retaining a value of 10 is that of both chop actions failing, with a likelihood of $.1 \times .1$. In any case, the controller advises a sensing action, which would return *down*, and so the plan terminates for $\alpha = [chop(1,0) \cdot getd \cdot chop(1,1) \cdot getd]$ with $Know(Bel(d < 10, now) > .9, do(\alpha, S_0))$.

Example 5.4. Consider the tree chop problem with noisy sensors and effectors. Suppose the initial theory and *alt*-axioms are (3) and (6) as before, but the likelihoods for the effector is given by (7) and that for the sensor is given by (4). Finally, suppose our goal is $Bel(d \leq 5, now) > .8$.

A plan that is epistemically correct for the goal is given in Figure 3 wrt observed values of 5.5, 4.5 and 3.9. We only argue for S_0. Assume also that $\{< 6, > 6\} \in O$ and that the numeric values obtained from the sensor map to these binary outcomes.

Here, the controller advises *chop*, after which the belief in $\psi = d \leq 5$ is $> .5$. This is because the likelihood of the action succeeding is more than it failing, and given the prior in $d \leq 5$ is .5, the posterior should be clearly greater than .5. Suppose now the sensed value is 5.5. The belief in ψ drops to $< .5$. The controller advises another *chop*, at which point the belief in ψ increases to $> .5$. Next, we observe a reading of 4.5 followed by 3.9. The controller terminates. On termination, it can be verified that the robot knows that the degree of belief in ψ is $> .8$.

In a noise-free setting, our definition of an epistemically correct plan is downward compatible with Definition 2.5 (and thus, Definition 4.2):

Theorem 5.5. *Suppose $\mathcal{D}$ is a noise-free action theory, $\mathcal{X}, \Sigma$ as above, and ϕ is any formula not mentioning K. If $\mathcal{X}$ is epistemically correct for ϕ, then it is correct for ϕ in the sense of Definition 2.5.*

Proof. Suppose $\mathcal{X}$ is epistemically correct but not correct. Then there is some s such that $K(s, S_0)$ and $\neg\exists s'' T^*(Q_0, s, Q_F, s'') \wedge \phi(s'')$. By assumption, $V^*(Q_0, \bar{s}, Q_F, \overline{s'}) \wedge \phi(\overline{s'})$ for some s'. Thinking of ⟨control states, situations⟩ are "nodes" in an execution path, the definition of V^* is the least set of pairs of nodes containing: $\langle Q_0, t\rangle$ for all situation terms t such that $K(t, s)$, which includes s because K is assumed to be an equivalence relation; $\langle q, do(a_1, t)\rangle$ for all situation terms t such that $K(t, s)$ provided Q_0 advises a_1, it is executable at every t and the sensing function returns o for a_1 at s and $\delta(Q_0, o) = q$; and so on. By assumption, V^* contains as node $\langle Q_F, s'\rangle$ for $s' = do(a_1 \cdots a_k, s)$. But, by the definition of T^*, it follows that $T^*(Q_0, s, Q_F, s')$. Moreover, since $\phi(\overline{s'})$ and K is reflexive, $\phi(s')$. Contradiction.

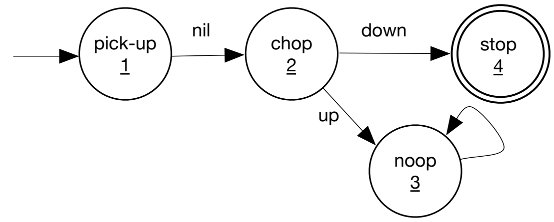

Figure 4: a problematic controller

In general, in the presence of noise, as one would expect, epistemic correctness diverges from correctness criteria based on plan evaluation at initial worlds. For example, we have:

Theorem 5.6. *Suppose $\mathcal{D}$ is any action theory, $\mathcal{X}, \Sigma$ as above, and ϕ is any formula not mentioning K. If $\mathcal{X}$ is epistemically correct for ϕ, then it does not follow that $\mathcal{X}$ satisfies* (8).

Proof. To prove this result, it suffices to provide a (possibly unwise) controller that is epistemically correct, but does not satisfy (8). Consider the tree chop problem for a tree of unit thickness, but with an additional action for picking up a saw. Suppose there is only a single initial world S_0, and $\forall a, s(Poss(a, s) \equiv true)$. Suppose the pick-up can fail, that is, suppose $alt(pickup, b, z) \equiv b = noop$, where *noop* is the action of doing nothing. Suppose neither of these actions provide the agent with any meaningful sensing result: that is, $\forall s\ SF(pickup, s) = SF(noop, s) = nil$. Imagine a controller like in Figure 4, where the control states are designated by numbers. (That is, q_1 is the control state advising *pickup* and q_4 is terminating state.) Initially, the controller advises *pickup*, leading to two successor situations: $do(pickup, S_0)$ and $do(noop, S_0)$. By way of the definition of SF and $Poss$ for these actions, we have $V^*(q_1, \overline{S_0}, q_2, \overline{do(pickup, S_0)})$, $U^*(q_1, S_0, q_2, do(pickup, S_0))$ and $U^*(q_1, S_0, q_2, do(noop, S_0))$. Suppose now $SF(chop, \overline{do(pickup, S_0)}) = down$ and so we have $V^*(q_1, \overline{S_0}, q_4, \overline{do(pickup \cdot chop, S_0)})$, and by construction, $[d(now) = 0](\overline{do(pickup \cdot chop, S_0)})$. However, suppose $SF(chop, do(noop, S_0)) = up$. Then we have $U^*(q_1, S_0, q_3, do(noop \cdot chop, S_0))$, which will not terminate.

The intuitive reason is that the termination conditions defined over U^* respond to sensing results along every execution path, whereas V^* only responds to the sensing outcomes for the path of advised actions from an initial world and how belief changes with it. This can be seen to be not surprising: among other things, the much stronger (8) is not needed because incompatible worlds will get discarded after sensing.

Let us conclude the section with two remarks. First, as mentioned earlier, the definition of V^* does not look very different from the semantics of noise-free belief-based planning, and this is good news: if the space of belief states is designed carefully to account for noise, algorithms for noise-free generalised planning may carry over to the stochastic case. Second, note that V^* was defined to include an explicit reference to sensing outcomes from the environment, which is external to the agent. A result in [21] shows that it is not possible to realise that we are making progress towards the goal without such a construction in belief-based planning.

6 DISCUSSION AND CONCLUSIONS

Generalising plans has been of interest since the early days of planning [27]. Algorithmic proposals to synthesise plans that generalise varied widely in methodology, ranging from interactive theorem proving [28] to learning from examples [29]. The convergence of these approaches to synthesise plans that solve multiple problem instances is a recent effort [4, 19, 30–32]. We refer interested readers to [31] for a comprehensive list of references, and [5, 33] for recent advances on handling nondeterminism. Outside of Levesque's account on the correctness of program-like plans as an epistemic formulation, there are numerous variants [2, 5, 33–35]. The semantics U^* extended Levesque's T^* to handle noisy acting, and V^* further extends that to noisy sensing, thereby obtaining a full generalisation of the formulation to handle nondeterminism.

Belief-based planning, which we touch upon, is widely studied, e.g., [24], and the usual approach is to formulate a nondeterministic (conformant) planning problem that treats belief states as first-class citizens. Our definition of V^* can be seen as a formalisation of this semantics against a logic of probabilistic belief and action. In that regard, the motivation behind this work is close in spirit to knowledge-based programs [36, 37] and its stochastic extension [26]. These are formulated using the situation calculus and the high-level programming language GOLOG, but, of course, variant languages are also popular for developing such planning accounts [38]. At the outset, there are significant reasons to develop a semantics customised to memoryless plans, as we argue below. Moreover, since there are a number of generalised planning algorithms that synthesise loopy plans [4], an execution semantics tailored to that representation is useful to understand how those algorithms can be applied to domains with noise.

Let us begin by observing that memoryless plans can be easily encoded as GOLOG programs consisting of atomic physical actions and branches based on sensing outcomes. In general, given a program δ, one is interested in showing that

$$\mathcal{D} \cup \Theta \models Do(\delta, S_0, do(\sigma, S_0))$$

where Θ encodes the single-step transition semantics of δ, and σ is a ground sequence of actions such that δ terminates in $do(\sigma, S_0)$. The key feature of knowledge-based programs is that δ can mention *Know*, and the probabilistic belief operator *Bel* in [26]. Nonetheless, note that if δ does not mention *Bel*, the entailment criteria above is weaker than Definition 2.5 as it only looks at S_0. But if it mentions the *Bel* operator, then it seems closer to Definition 5.2, but at the cost of a more cumbersome plan structure: δ can have unbounded memory (via while loops), can refer to complicated state properties, and is subjective, whereas Definition 5.2 is defined for memoryless plans built purely from a finite set of atomic actions.

So, in the current paper, the end result is an account of correctness with widely-studied loopy plan structures, which eschews the complications of GOLOG but is able to achieve almost as much. Naturally, then, relating knowledge-based programs and loopy plans (in belief-based settings) is likely to be of considerable theoretical interest, as would a closer study of the two execution semantics. (Cf. also [34] on memoryless structures being effective, and [37] on knowing how to execute GOLOG programs.)

Despite focusing on probabilities and nondeterminism, this paper has established no connection to the large body of work on Markov decision processes [39]. Mostly, decision-theoretic planning frameworks are characterised in terms of optimality criteria against expected rewards, often enabled via dynamic programming, while we have treated goals as arbitrary formulas that are to be satisfied, as would symbolic planning frameworks such as [2]. Nonetheless, one can imagine ways of recasting expected rewards in terms of goal satisfaction or vice versa [40], and that is arguably worth doing in the context of this paper so as to relate to efforts such as [41]. Interestingly, recent robotics planners such as [16] eschews a planning paradigm that advises actions for every belief state, as one would in partially observable Markov decision processes, and instead resorts to a scheme that computes plans for a designated initial belief state, as in belief-based planning. Moreover, as mentioned before, ultimately the goal here was to generalise Levesque's account and to provide a rigorous foundation for extensions such as the handling of non-unique prior distributions.

As a final remark, like in Levesque's original formulation, one can motivate a generic planning procedure as follows:

input: ϕ, $E^* \in \{T^*, U^*, V^*\}$, Δ is a correctness criteria
repeat with $\mathcal{X} \in$ FINITE STATE CONTROLLERS
if $\mathcal{D} \cup \Sigma \models \forall s.\ K(s, S_0) \supset \Delta(E^*, \phi, s)$ **then return** $\mathcal{X}$

Naturally, we do not expect to use a full-blown logical framework for planning, nor do we expect planners to actually use such a procedure in practise. Languages like the ones in [16, 42] seem entirely reasonable. It is also conceivable that existing algorithms for generalised planning, such as bounded AND/OR searches, can be adapted for stochastic settings, as argued earlier, possibly by leveraging abstraction techniques [5, 33]. We hope this paper is also useful for approaching and resolving that line of inquiry.

REFERENCES

[1] H. J. Levesque. What is planning in the presence of sensing? In *Proc. AAAI / IAAI*, pages 1139–1146, 1996.
[2] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1–2):35 – 84, 2003.
[3] B. Bonet, H. Palacios, and H. Geffner. Automatic derivation of memoryless policies and finite-state controllers using classical planners. In *ICAPS*, 2009.
[4] Y. Hu and G. De Giacomo. A generic technique for synthesizing bounded finite-state controllers. In *ICAPS*, 2013.
[5] S. Srivastava, S. Zilberstein, A. Gupta, P. Abbeel, and S. Russell. Tractability of planning with loops. In *AAAI*, 2015.
[6] F. Kominis and H. Geffner. Beliefs in multiagent planning: From one agent to many. In *ICAPS*, pages 147–155, 2015.
[7] C. Muise, V. Belle, P. Felli, S. McIlraith, T. Miller, A. Pearce, and L. Sonenberg. Planning over multi-agent epistemic states: A classical planning approach. In *Proc. AAAI*, 2015.
[8] M. J. Matarić. *The robotics primer*. Mit Press, 2007.
[9] R. B. Scherl and H. J. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2):1–39, 2003.
[10] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[11] F. Bacchus, J. Y. Halpern, and H. J. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence*, 111(1–2):171 – 208, 1999.
[12] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press, 1990.
[13] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994.
[14] B.P. Kooi. Probabilistic dynamic epistemic logic. *Journal of Logic, Language and Information*, 12(4):381–408, 2003.
[15] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99 – 134, 1998.
[16] L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *I. J. Robotic Res.*, 32(9-10):1194–1227, 2013.
[17] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463–502, 1969.
[18] R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.
[19] Y. Hu and H. J. Levesque. A correctness result for reasoning about one-dimensional planning problems. In *IJCAI*, pages 2638–2643, 2011.
[20] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
[21] S. Sardiña, G. De Giacomo, Y. Lespérance, and H. J. Levesque. On the limits of planning over belief states under strict uncertainty. In *KR*, pages 463–471, 2006.
[22] T. Siméon, J. Laumond, J. Cortés, and A. Sahbani. Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):729–746, 2004.
[23] R. A. Knepper and M. T. Mason. *Realtime Informed Path Sampling for Motion Planning Search*, pages 401–417. Springer International Publishing, Cham, 2017.
[24] B. Bonet and H. Geffner. Planning with incomplete information as heuristic search in belief space. In *AIPS*, pages 52–61, 2000.
[25] R.P.A. Petrick and F. Bacchus. Extending the knowledge-based approach to planning with incomplete information and sensing. In *Proc. ICAPS*, pages 2–11, 2004.
[26] V. Belle and H. J. Levesque. Allegro: Belief-based programming in stochastic dynamical domains. In *IJCAI*, 2015.
[27] R.E. Fikes, P.E. Hart, and N.J. Nilsson. Learning and executing generalized robot plans. *Artificial intelligence*, 3:251–288, 1972.
[28] W. Stephan and S. Biundo. Deduction-based refinement planning. In *AIPS*, pages 213–220, 1996.
[29] E. Winner and M. M. Veloso. LoopDISTILL: Learning domain-specific planners from example plans. In *Workshop on AI Planning and Learning, ICAPS*, 2007.
[30] B. Bonet, H. Palacios, and H. Geffner. Automatic derivation of finite-state machines for behavior control. In *AAAI*, 2010.
[31] S. Srivastava. *Foundations and Applications of Generalized Planning*. PhD thesis, Department of Computer Science, University of Massachusetts Amherst, 2010.
[32] H.J. Levesque. Planning with loops. In *Proc. IJCAI*, pages 509–515, 2005.
[33] B. Bonet, G. De Giacomo, H. Geffner, and S. Rubin. Generalized planning: Non-deterministic abstractions and trajectory constraints. In *IJCAI*, pages 873–879, 2017.
[34] F. Lin and H. J. Levesque. What robots can do: Robot programs and effective achievability. *Artif. Intell.*, 101(1-2):201–226, 1998.
[35] V. Belle and H. Levesque. Foundations for generalized planning in unbounded stochastic domains. In *KR*, 2016.
[36] R. Reiter. On knowledge-based programming with sensing in the situation calculus. *ACM Trans. Comput. Log.*, 2(4):433–457, 2001.
[37] Y. Lespérance, H. J. Levesque, F. Lin, and R. B. Scherl. Ability and knowing how in the situation calculus. *Studia Logica*, 66(1):165–186, 2000.
[38] Y. Martin and M. Thielscher. Integrating reasoning about actions and Bayesian networks. In *International Conference on Agents and Artificial Intelligence*, Valencia, Spain, January 2009.
[39] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.
[40] H. Geffner and B. Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Morgan and Claypool Publishers, 2013.
[41] P. Poupart and C. Boutilier. Bounded finite state controllers. In *NIPS*, pages 823–830, 2004.
[42] S. Sanner and K. Kersting. Symbolic dynamic programming for first-order pomdps. In *Proc. AAAI*, pages 1140–1146, 2010.