Regression and Progression in Stochastic Domains

Citation for published version:
Belle, V & Levesque, HJ 2020, 'Regression and Progression in Stochastic Domains', *Artificial Intelligence*, vol. 281, 103247. https://doi.org/10.1016/j.artint.2020.103247

Digital Object Identifier (DOI):
10.1016/j.artint.2020.103247

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Artificial Intelligence

Regression and Progression in Stochastic Domains☆

Vaishak Belle[a,b,*], Hector J. Levesque[c]

[a] *School of Informatics, University of Edinburgh, Edinburgh, UK.*
[b] *Alan Turing Institute, London, UK.*
[c] *Department of Computer Science, University of Toronto, Toronto, Canada.*

Abstract

Reasoning about degrees of belief in uncertain dynamic worlds is fundamental to many applications, such as robotics and planning, where actions modify state properties and sensors provide measurements, both of which are prone to noise. With the exception of limited cases such as Gaussian processes over linear phenomena, belief state evolution can be complex and hard to reason with in a general way, especially when the agent has to deal with categorical assertions, incomplete information such as disjunctive knowledge, as well as probabilistic knowledge. Among the many approaches for reasoning about degrees of belief in the presence of noisy sensing and acting, the logical account proposed by Bacchus, Halpern, and Levesque is perhaps the most expressive, allowing for such belief states to be expressed naturally as constraints. While that proposal is powerful, the task of how to plan effectively is not addressed. In fact, at a more fundamental level, the task of projection, that of reasoning about beliefs effectively after acting and sensing, is left entirely open.

To aid planning algorithms, we study the projection problem in this work. In the reasoning about actions literature, there are two main solutions to projection: regression and progression. Both of these have proven enormously useful for the design of logical agents, essentially paving the way for cognitive robotics. Roughly, regression reduces a query about the future to a query about the initial state. Progression, on the other hand, changes the initial state according to the effects of each action and then checks whether the formula holds in the updated state. In this work, we show how both of these generalize in the presence of degrees of belief, noisy acting and sensing. Our results allow for both discrete and continuous probability distributions to be used in the specification of beliefs and dynamics.

Keywords: Knowledge representation, Reasoning about action, Reasoning about knowledge, Reasoning about uncertainty, Cognitive robotics

1. Introduction

Reasoning about *degrees of belief* in uncertain dynamic worlds is fundamental to many applications, such as robotics and planning, where actions modify state properties and sensors provide measurements, both of which are prone to noise. However, there seem to be two disparate paradigms to address this concern, both of which have their limitations. At one extreme, there are logical formalisms, such as the *situation calculus* [51, 58], which allows us to express strict uncertainty, and exploits regularities in the effects actions have on propositions to describe physical laws compactly. Probabilistic sensor fusion, however, has received less attention here. (Notable exceptions will be discussed in the penultimate section.) At the other extreme, revising beliefs after noisy observations over rich error profiles is effortlessly addressed using *probabilistic techniques* such as Kalman filtering and Dynamic Bayesian Networks [20, 21]. However, in these frameworks, a complete specification of the dependencies between variables is taken as given, making it difficult to deal with other forms of incomplete knowledge as well as complex actions that shift dependencies between variables in nontrivial ways.

☆A preliminary version of this work has appeared in [7, 8]. In particular, the results on regression were first reported in [7], but that account was limited to noise-free actions only. The results on progression were first reported in [8].

*Corresponding author. Vaishak Belle was partly supported by a Royal Society University Research Fellowship.
Email addresses: vaishak@ed.ac.uk (Vaishak Belle), hector@cs.toronto.edu (Hector J. Levesque)

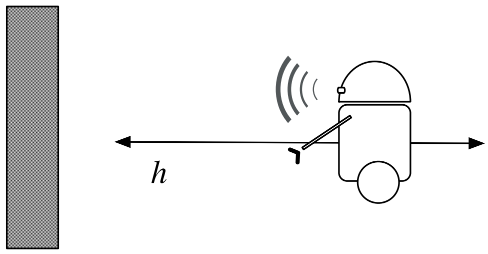

Figure 1: Robot moving towards a wall.

An influential but nevertheless simple proposal by Bacchus, Halpern and Levesque [2], BHL henceforth, was among the first to merge these broad areas in a general way. Their specification is widely applicable because it is not constrained to particular structural assumptions. In a nutshell, they extend the situation calculus language with a provision for specifying the degrees of belief in formulas in the initial state, closely fashioned after intuitions on incorporating probability in modal logics [34, 26]. This then allows incomplete and partial specifications, which might be compatible with one or very many initial distributions and sets of independence assumptions, with beliefs following at a corresponding level of specificity. Moreover, together with a rich action theory, the model not only exhibits Bayesian conditioning [55] (which, then, captures special cases such as Kalman filtering [69]), but also allows flexibility in the ways dependencies and distributions may change over actions.

While that proposal is powerful, the task of how to plan effectively is not addressed. In essence, this would correspond to a flavor of epistemic planning [3] where the state of knowledge, actions and sensing are mixtures of logical and probabilistic assertions. In fact, at a more fundamental level, the task of projection, that of reasoning about beliefs effectively after acting and sensing, is left entirely open. More precisely, while changing degrees of belief do indeed emerge as *logical entailments* of the given action theory, no procedure is given for computing these entailments. On closer examination, in fact, this is a two-part question:

(i) How do we effectively reason about beliefs in a particular state?

(ii) How do we effectively reason about belief state evolution and belief change?

In the simplest case, part (i) puts aside acting and sensing, and considers reasoning about the *initial* state only, which is then the classical problem of (first-order) probabilistic inference. We do not attempt to do a full survey here, but this has received a lot of attention [56, 32, 18, 11].

This work is about part (ii). Addressing this concern would not only aid planning algorithms, but also has a critical bearing on the assumptions made about the domain for tractability purposes. For example, if the initial state supports a decomposed representation of the distribution, can we expect the same after actions? In the exception of very limited cases such as Kalman filtering that harness the conjugate property of Gaussian processes, the situation is discouraging. In fact, even in the slightly more general case of Dynamic Bayesian Networks, which are in essence atomic propositions, if one were to assume that state variables are independent at time 0, they can become fully correlated after a few steps [19, 17, 33]. Dealing with complex actions, incomplete specifications and mixed representations, therefore, is significantly more involved.

In the reasoning about actions literature, where the focus is on qualitative (non-probabilistic) knowledge, there are two main solutions to projection: regression and progression [58]. Both of these have proven enormously useful for the design of logical agents, essentially paving the way for *cognitive robotics* [42]. Roughly, regression reduces a query about the future to a query about the initial state. Progression, on the other hand, changes the initial state according to the effects of each action and then checks whether the formula holds in the updated state. In this work, we show how both of these generalize in the presence of degrees of belief, noisy acting and sensing. Our results allow for both discrete and continuous probability distributions to be used in the specification of beliefs and dynamics, that leverage a recent extension of the BHL framework to mixed discrete-continuous domains [10].

To elaborate on the regression result, we show that it is general, not requiring (but allowing) structural constraints about the domain, nor imposing (but allowing) limitations to the family of actions. Regression derives a mathematical formula, using term and formula *substitution* only, that relates belief after a sequence of actions and observations, even when they are noisy, to beliefs about the initial state. That is, among other things, if the initial state supports efficient factorizations, regression will maintain this advantage no matter how actions affect the dependencies between state variables over time. Going further, the formalism will work seamlessly with discrete probability distributions, probability densities, and perhaps most significantly, with difficult combinations of the two.

To see a simple example of what goal regression does, imagine a robot facing a wall and at a certain distance h to it, as in Figure 1. The robot might initially believe h to be drawn from a uniform distribution on $[2, 12]$. Assume the robot moves away by 2 units and is now interested in the belief about $h \leq 5$. Regression would tell the robot that this is equivalent to its initial beliefs about $h \leq 3$ which here would lead to a value of .1. To see a nontrivial example, imagine now the robot is also equipped with a sonar unit aimed at the wall, that adds Gaussian noise with mean μ and variance σ^2. After moving away by 2 units, if the sonar were now to provide a reading of 8, then regression would derive that belief about $h \leq 5$ is equivalent to

$$\frac{1}{\gamma} \int_2^3 .1 \times \mathcal{N}(6 - x; \mu, \sigma^2) \, \mathrm{d}x.$$

where γ is the normalization factor. Essentially, the posterior belief about $h \leq 5$ is reformulated as the product of the prior belief about $h \leq 3$ and the likelihood of $h \leq 3$ given an observation of 6. (That is, observing 8 after moving away by 2 units is related here to observing 6 initially.)

We believe the broader implications of this result are two-fold. On the one hand, as we show later, simple cases of belief state evolution, as applicable to conjugate distributions for example, are special cases of regression's backward chaining procedure. Thus, regression could serve as a *formal basis* to study probabilistic belief change wrt limited forms of actions. On the other hand, our contribution can be viewed as a methodology for combining actions with recent advances in probabilistic inference, because reasoning about actions reduces to reasoning about the initial state.

To elaborate on the progression result, it has been argued that for long-lived agents like robots, continually updating the current view of the state of the world, is perhaps better suited. Lin and Reiter [47] show that progression is always second-order definable, and in general, it appears that second-order logic is unavoidable [76]. However, Lin and Reiter also identify some first-order definable cases by syntactically restricting situation calculus basic action theories, and since then, a number of other special cases have been studied [48].

While Lin and Reiter intended their work to be used on robots, one criticism leveled at their work, and indeed at much of the work in cognitive robotics, is that the theory is far removed from the kind of continuous uncertainty and noise seen in typical robotic applications. What exactly filtering mechanisms (such as Kalman filters) have to do with Lin and Reiter's progression has gone unanswered, although it has long been suspected that the two are related.

Our result remedies this situation. However, as we discuss later, progression in stochastic domains is complicated by the fact that actions can transform a continuous distribution to a mixed one. To obtain a closed-form result, we introduce a property of basic action theories called *invertibility*, closely related to invertible functions in real analysis [70]. We identify syntactic restrictions on basic action theories that guarantee invertibility. For our central result, we show a first-order *progression* of degrees of belief against noise in effectors and sensors for action theories that are invertible.

We structure this article as follows. In the preliminaries section, we cover the situation calculus, recap BHL and go over the essentials of its continuous extension. (This is taken with slight modifications from [10].) We then present regression for discrete domains, followed by regression for general domains. We then turn to a few special cases, such as conjugate distributions. Next, we turn to invertible theories, and discuss progression. Finally, we conclude after discussing related work.

2. Background

We work with the language $\mathcal{L}$ of the situation calculus [51], as developed in [58]. It is a special-purpose knowledge representation formalism for reasoning about dynamical systems. The formalism is best understood by arranging the world in terms of three kinds of things: *situations*, *actions* and *objects*. Situations represent "snapshots," and can be viewed as possible histories. A set of initial situations correspond to the ways the world can be prior to the occurrence of actions. The result of doing an action, then, leads to a successor (non-initial) situation. Naturally, dynamic worlds change the properties of objects, which are captured using predicates and functions whose last argument is always a situation, called *fluents*.

Logical Language

Formally, the language $\mathcal{L}$ of the situation calculus is a many-sorted dialect of predicate calculus, with sorts for *actions*, *situations* and *objects* (for everything else). (We do not review standard predicate logic here; see, for example, [25, 65]. We further assume familiarity with the notions of *models*, *structures*, *satisfaction* and *entailment*.) In full length, let $\mathcal{L}$ include:

- logical connectives $\neg, \forall, \wedge, =$, with other connectives such as $\supset$ understood for their usual abbreviations;
- an infinite supply of *variables* of each sort;
- an infinite supply of *constant* symbols of the sort object;
- for each $k \geq 1$, *object function* symbols $g_1, g_2, \ldots$ of type $(action \cup object)^k \rightarrow object$;
- for each $k \geq 0$, *action function* symbols $A_1, A_2, \ldots$ of type $(action \cup object)^k \rightarrow action$;
- a special *situation function* symbol *do*: $action \times situation \rightarrow situation$;
- a special predicate symbol *Poss*: $action \times situation$;[1]
- for each $k \geq 0$, *fluent function* symbols $f_1, f_2, \ldots$ of type $(action \cup object)^k \times situation \rightarrow object$;
- a special constant S_0 to represent the actual initial situation.

To reiterate, apart from some syntactic particulars, the logical basis for the situation calculus is the regular (many-sorted) predicate calculus.[2] So, terms and well-formed formulas are defined inductively, as usual, respecting sorts. See, for example, [58] for an exposition.

Dynamic worlds are enabled by performing actions, and in the language, this is realized using the *do* operator. That is, the result of doing an action a at situation s is the situation $do(a, s)$. Functional fluents, which take situations as arguments, may then have different values at different situations, thereby capturing *changing properties* of the world. As noted, the constant S_0 is assumed to give the actual initial state of the domain, but the agent may consider others possible that capture the beliefs and ignorance of the agent. In general, we say a situation is an *initial* one when it is a situation without a *predecessor*:

$$Init(s) \doteq \neg\exists a, s'.\ s = do(a, s').$$

The picture that emerges is that situations can be structured as a set of trees, each rooted at an initial situation and whose edges are actions. We use ι to range over such initial situations only, and let δ denote sequences of action terms or variables, and freely use this with *do*, that is, if $\delta = [a_1, \ldots, a_n]$ then $do(\delta, s)$ stands for $do(a_n, do(\ldots, do(a_1, s)\ldots))$.

Domains are modeled in the situation calculus as *axioms*. A set of $\mathcal{L}$-sentences specify the actions available, what they depend on, and the ways they affect the world. Specifically, these axioms are given in the form of a *basic action theory* [58], reviewed shortly, that appeal to the formulation of successor state axioms, which incorporates a monotonic solution to the frame problem.

We follow three notational conventions. We often suppress the situation argument in a formula ϕ, or use a distinguished variable *now*. Either way, $\phi[t]$ is used to denote the formula with that variable replaced by t, *e.g.* both $(f < 12)[s]$ and $(f(now) < 12)[s]$ mean $f(s) < 12$. We use conditional *if-then-else* expressions in formulas throughout. We write $f = \text{IF } \phi \text{ THEN } t_1 \text{ ELSE } t_2$ to mean $[\phi \wedge f = t_1] \vee [\neg\phi \wedge f = t_2]$. In case quantifiers appear inside the *if*-condition, we take some liberties with notation and the scope of variables in that we write $f = \text{IF } \exists x.\ \phi \text{ THEN } t_1 \text{ ELSE } t_2$ to mean $\exists x\ [\phi \wedge f = t_1] \vee [(f = t_2) \wedge \neg\exists x.\ \phi]$. Finally, it is also useful to have

[1] We will subsequently introduce a few more distinguished predicates when modeling knowledge, sensing and nondeterminism.

[2] For simplicity, only functional fluents are introduced, and their predicate counterparts are ignored. (Distinguished symbols like *Poss* are an exception.) This is without loss of any generality since predicates can be thought of functions that take one of two values, the first denoting *true* and the other denoting *false*.

certain abbreviations that will macro expand to $\mathcal{L}$-formulas. These abbreviations are used as logical terms, that is, as arguments to functions or predicates. If E is such an expression, its expansion is characterized by a definition of the form $E = u \doteq \phi(u)$, where u is a variable, and $\phi(u)$ is an $\mathcal{L}$-formula with u free. This should be interpreted as follows. Let $\rho(u)$ be any atomic formula with u free. Then the expression $\rho(E)$ should be understood as standing for the formula $\exists u(\phi(u) \wedge \rho(u))$.

Basic action theory

Following [58], we model dynamic domains in $\mathcal{L}$ by means of a *basic action theory* $\mathcal{D}$, which consists of domain-independent *foundational* axioms, *unique name* axioms for actions (see [58]), and (1) axioms $\mathcal{D}_0$ that describe what is true in the initial states, including S_0;[3] (2) precondition axioms[4] of the form $Poss(A(\vec{x}), s) \equiv \Pi_A(\vec{x}, s)$ describing executability conditions using a special fluent *Poss*; and (3) successor state axioms of the following form stipulating how fluents change:

$$f(do(a, s)) = u \equiv \Phi_f(u, a, s).$$

For example, consider the action $fwd(z)$ of moving precisely z units towards the wall, but the motion stops when the wall is reached:

$$\begin{aligned} h(do(a, s)) = u \equiv\ & \exists z[a = fwd(z) \wedge u = \max(0, h(s) - z)] \vee \\ & \neg\exists z[a = fwd(z)] \wedge u = h(s). \end{aligned}$$

Moving away from the wall can be accomplished by providing a negative number as an argument to the action. This sentence also states that $fwd(z)$ is the only action affecting fluent h, in effect incorporating a solution to the frame problem [58]. This successor state axiom can also be written using *if-then-else* as:

$$\begin{aligned} h(do(a, s)) = \ & \text{IF } \exists z(a = fwd(z)) \\ & \text{THEN } \max(0, h(s) - z) \text{ ELSE } h(s). \end{aligned} \tag{1}$$

(If there are more actions that affect this fluent, they can be nested within *else*.) Henceforth, successor state axioms are taken to be in the form:

$$f(do(a, s)) = E_f(a)[s]. \tag{2}$$

Thus, as a consequence of our notational conventions introduced earlier, we will be able to write successor state axioms as equality-expressions, and use the RHS as terms than can be substituted in formulas.

Given an action theory, an agent reasons about actions by means of entailments of $\mathcal{D}$. A fundamental task in reasoning about action is that of *projection* [58], where we test which properties hold after actions. Formally, suppose ϕ is a situation-suppressed formula or uses the special symbol *now*. Given a sequence of actions a_1 through a_n, we are often interested in asking whether ϕ holds after these:

$$\mathcal{D} \models \phi[do([a_1, \ldots, a_n], S_0)]?$$

Entailments are wrt standard Tarskian models, but we will also assume that models assign the usual interpretations to $=, <, >, 0, 1, +, \times, /, -, e, \pi$, and x^y (exponentials). See [10] for discussions.

Likelihood and belief

The BHL model of belief [2] builds on a treatment of *knowledge* in $\mathcal{L}$ [62]. Here we present a simpler variant based on two distinguished fluents l and p [10].

The term $l(a, s)$ is intended to denote the likelihood of action a in situation s. For example, suppose $sonar(z)$ is the action of reading the value z from a sonar that measures the distance to the wall, h. We might assume that this action is characterized by a simple discrete error model:

$$l(sonar(z), s) = \text{IF } |h(s) - z| \leq 1 \text{ THEN } 1/3 \text{ ELSE } 0 \tag{3}$$

[3] Note that $\mathcal{D}_0$ can include any (classical) first-order sentence about S_0, such as $h(S_0) > 12$ and $f_1(S_0) \neq 2 \vee f_2(S_0) = 5$.

[4] Free variables in any of these axioms should be understood as universally quantified from the outside.

which stipulates that the difference between a reading of z and the true value h is either $\{0, -1, 1\}$ with probability $1/3$, assuming that h and z take integer values.

To capture the idea that the likelihood of reading z is obtained from a normal curve whose mean is h, one would use:

$$l(sonar(z), s) = \mathcal{N}(z; h, 4)[s] \tag{4}$$

which gives the sensor's reading a variance of 4. This is referred to as an additive Gaussian noise model [69].[5]

Note that the l-axioms are more expressive than those typically seen in probabilistic formalisms, and can be context-dependent. For example, to model a sensor with systematic bias at subzero temperatures, we might have

$$\begin{aligned} l(sonar(z), s) = {} & \text{IF } temp(s) > 0 \\ & \text{THEN } \mathcal{N}(z; h, 1) \text{ ELSE } \mathcal{N}(z; h + 2, 1). \end{aligned}$$

In general, the action theory $\mathcal{D}$ is assumed to contain for each sensor $sense(\vec{x})$ that measures a fluent f, an axiom of the form:

$$l(sense(\vec{x}), s) = Err_{sense}(\vec{x}, f(s)),$$

where $Err_{sense}(u_1, u_2)$ is some expression with only two free variables u_1 and u_2, both numeric.[6] Noise-free physical actions are given a likelihood of 1. (Noisy physical actions will be treated in a subsequent section.)

Next, the p fluent determines a (subjective) probability distribution on situations.[7] The term $p(s', s)$ denotes the relative *weight* accorded to situation s' when the agent happens to be in situation s, as in modal probability logics [26]. Now, the task of the modeler is to specify the initial properties of p as part of $\mathcal{D}_0$ using ι and S_0, *e.g.*:

$$p(\iota, S_0) = \text{ IF } h(\iota) \in \{2, \ldots, 11\} \text{ THEN } .1 \text{ ELSE } 0 \tag{5}$$

says that h is drawn from a uniform distribution, corresponding to Figure 1. Suppose we are instead in a setting where the robot can move in a 2-dimensional space, as shown in Figure 2. Suppose the fluent v captures it position along the Y-axis. Then a constraint such as

$$p(\iota, S_0) = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; 0, 1)[\iota] \tag{6}$$

says that the agent does not know the initial values of h and v, but thinks of them as drawn independently from a uniform distribution on [2, 12], and from a standard normal distribution. Since p is just like any other fluent, the framework is more expressive than many probabilistic formalisms. For example,

$$\begin{gathered} \forall\iota(p(\iota, S_0) = \mathcal{U}(h; 2, 3)[\iota]) \\ \vee \\ \forall\iota(p(\iota, S_0) = \mathcal{U}(h; 10, 20)[\iota]) \end{gathered} \tag{7}$$

says that the agent believes h to be uniformly distributed on [2,3] or on [10,20], without being able to say which.

To give p the intended properties, the following non-negative constraint is assumed to be included in $\mathcal{D}_0$ [2]:

$$\forall\iota, s.\, p(s, \iota) \geq 0 \wedge (p(s, \iota) > 0 \supset Init(s)) \tag{P1}$$

[5] Note that the BHL scheme is limited to discrete distributions, but we will shortly discuss its extension to continuous domains, and so are introducing the features of the full language.

[6] This captures the idea that the error model of a sensor measuring f depends only on the true value of f, and is independent of other factors. In a sense this follows the Bayesian model that conditioning on a random variable f is the same as conditioning on the event of observing f. But this is not required in general in the BHL scheme, an issue we ignore for this paper. See [2] for discussions. Moreover, as usual [62], we assume that physical actions have trivial sensing values, and that sensing actions do not affect physical properties. Physical actions with non-trivial sensing axioms are to be treated as two separate actions, the first capturing the physical effects and the second capturing the sensing ones.

[7] The p fluent is a numeric version of the K fluent used in modeling knowledge in the situation calculus [62]. (One could let $K(s', s)$ be an abbreviation for $p(s', s) > 0$, for example.) See Scherl and Levesque [62] on how features such as *positive* and *negative introspection* can be enabled by constraining the accessibility relation K, in a manner entirely analogous to standard modal logic [27]. (Different from standard modal logics [27], however, worlds are reified as part of the syntax, but this is a minor technicality.) We will not delve into these issues further here, and refer readers to works such as [5] that study properties of knowledge in the BHL scheme in more detail.

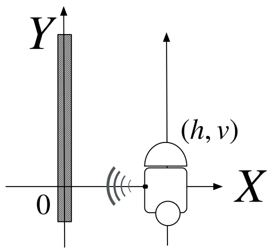

Figure 2: Robot moving along both axes.

Then, by means of a remarkably simple successor state axiom for p, (P2) below, the formal specification is complete.

$$\begin{aligned} & p(s', do(a,s)) = \\ & \qquad \text{IF } \exists s''.\ s' = do(a,s'') \land Poss(a,s'') \\ & \qquad\quad \text{THEN } p(s'',s) \times l(a,s'') \\ & \qquad \text{ELSE } 0 \end{aligned} \tag{P2}$$

In particular, the *degree of belief* in a formula ϕ can be accounted for in terms of an abbreviation:

$$Bel(\phi, s) \doteq \frac{1}{\gamma} \sum_{\{s' : \phi[s']\}} p(s', s) \tag{B}$$

where γ, the normalization factor, is understood throughout as the same expression as the numerator but with ϕ replaced by *true*, *e.g.* here γ is $\sum_{s'} p(s', s)$. Summation, and later integration, is to be understood as a logical term, so the expression above expands to a well-formed $\mathcal{L}$-formula, as shown in [10]. Note also that p and l are stipulated to define probability distributions; see [2] for discussions.

So, as in probability logics [26], belief is simply the total weight of worlds satisfying ϕ. But the novelty here is that in a dynamical setting, belief change via (B) is identical to Bayesian conditioning:

Proposition 1. *[10] Suppose $\mathcal{D}$ includes* (P1), (P2) *and the likelihood axiom for a sensor $sense(z)$ measuring f. Then*
$$\mathcal{D} \models Bel(f = t, do(sense(z), S_0)) = \frac{Bel(f = t, S_0) \cdot Err(z,t)}{\sum_x Bel(f = x, S_0) \cdot Err(z,x)}$$

Essentially, if the robot's sensors are informative, in the sense of returning values closer to the true value, beliefs are strengthened over time.

From sums to integrals

While the definition of belief in BHL has many desirable properties, it is defined in terms of a *summation* over situations, and therefore precludes fluents whose values range over the reals. The continuous analogue of (B) then requires *integrating* over some suitable space of values.

As it turns out, a suitable space can be found. First, assume that there are n fluents $f_1, \ldots, f_n$ in $\mathcal{L}$, and that these take no arguments other than the situation argument.[8] Next, suppose that that there is exactly one initial situation for every possible value of these fluents [45]:

$$[\forall \vec{x} \exists \iota \bigwedge f_i(\iota) = x_i] \land [\forall \iota, \iota'. \bigwedge f_i(\iota) = f_i(\iota') \supset \iota = \iota'] \tag{P3}$$

[8] As we shall shortly see, our reformulation of the BHL scheme requires us to enumerate the n random variables of the domain (for some large n) [10]. If we were to assume that the arguments of all fluents, even k-ary ones, are taken from finite sets then too we would be able to enumerate the n random variables of the domain. However, from the point of view of situation calculus basic action theories, fluents are typically allowed to take arguments from any set, including infinite ones. In probabilistic terms, this would correspond to having a joint probability distribution over infinitely many, perhaps uncountably many, random variables. We have as yet no good ideas about how to deal with it. See [10] for discussions.

Under these assumptions, it can be shown that the summation over all situations in (B) can be recast as a summation over all possible initial values $x_1, \ldots, x_n$ for the fluents:

$$Bel(\phi, s) \doteq \frac{1}{\gamma} \sum_{\vec{x}} P(\vec{x}, \phi, s) \tag{B$'$}$$

where $P(\vec{x}, \phi, s)$ is the (unnormalized) weight accorded to the *successor* of an initial world where f_i equals x_i:

$$\begin{aligned} &P(\vec{x}, \phi, do(\delta, S_0)) \doteq \\ &\qquad \text{IF } \exists\iota.\ \textstyle\bigwedge f_i(\iota) = x_i \land \phi[do(\delta, \iota)] \\ &\qquad\quad \text{THEN } p(do(\delta, \iota), do(\delta, S_0)) \\ &\qquad \text{ELSE } 0 \end{aligned}$$

for any action sequence δ. In a nutshell, because every situation has an initial situation as an ancestor, and because there is a bijection between initial situations and possible fluent values, it is sufficient to sum over fluent values to obtain the belief even for non-initial situations. Note that unlike (B), this one expects the final situation term $do(\delta, S_0)$ mentioning what actions and observations took place to be explicitly specified, but that is just what one expects when the agent reasons about its belief after acting and sensing.

The generalization to the continuous case then proceeds as follows. First, we observe that some (though possibly not all) fluents will be real-valued, and that $p(s', s)$ will now be a measure of *density* not weight. For example, if h is real-valued, we might have the following analogue to (5):

$$p(\iota, S_0) = \text{IF } 2 \leq h(\iota) \leq 12 \text{ THEN } .1 \text{ ELSE } 0 \tag{8}$$

which says that the true initial value of h is drawn from a uniform distribution on [2,12]. Similarly, the P term above now measures (unnormalized) density rather than weight.

Now suppose fluents are partitioned into two groups: the first k take their values $x_1, \ldots, x_k$ from $\mathbb{R}$, while the rest take their values $y_{k+1}, \ldots, y_n$ from countable domains, then the *degree of belief* in ϕ is an abbreviation for:

$$Bel(\phi, s) \doteq \frac{1}{\gamma} \int_{\vec{x}} \sum_{\vec{y}} P(\vec{x} \cdot \vec{y}, \phi, s) \tag{B$^+$}$$

That is, the belief in ϕ is obtained by ranging over all possible fluent values, and integrating and summing the densities of *successor* situations where ϕ holds. For ease of presentation, we lump the discrete and continuous variables as $\vec{x}$ and use the integral symbol to mean integration over uncountable domains or summation over countable domains as appropriate.

To summarize the formalization, a basic action theory $\mathcal{D}$ henceforth is assumed to additionally include: (a) (P1) and (P3) as part of $\mathcal{D}_0$; (b) (P2) as part of $\mathcal{D}$'s successor state axioms, and (c) sensor likelihood axioms.

3. Regression for Discrete Domains

We now investigate a computational mechanism for reasoning about beliefs after a trajectory. This is a generalization of the regression operator for knowledge over exact acting and sensing investigated in [62]. In this section, we focus on discrete domains, where a *weight*-based notion of belief would be appropriate. Domains with both discrete and continuous variables are reserved for the next section.

Formally, given a basic action theory $\mathcal{D}$, a sequence of actions δ, we might want to determine whether a formula ϕ *holds* after executing δ starting from S_0:

$$\mathcal{D} \models \phi[do(\delta, S_0)] \tag{9}$$

which is called *projection* [58]. When it comes to beliefs, and in particular how that changes after acting and sensing, we might be interested in *calculating* the degrees of belief in ϕ after δ: find a real number r such that

$$\mathcal{D} \models Bel(\phi, do(\delta, S_0)) = r. \tag{10}$$

The obvious method for answering (9) is to translate both $\mathcal{D}$ and ϕ into a predicate logic formula. This approach, however, presents a serious computational problem because belief formulas expand into a large number of sentences using (P2), resulting in an enormous search space with initial and successor situations. The other issue with this approach is that sums (and integrals in the continuous case) reduce to complicated second-order formulas.

We now introduce a *regression* procedure to simplify both (9) and (10) to queries about $Bel(\phi, S_0)$, over arithmetic expressions, for which standard probabilistic reasoning methods can be applied. For this purpose, in the sequel, *Bel* is treated as a *special syntactic operator* rather than as an abbreviation for other formulas. To see a simple example of the procedure, imagine the robot is interested in the probability of $h = 7$, given (5), after reading 5 from a sonar:

$$Bel(h = 7, do(sonar(5), S_0)) \tag{11}$$

If we are to take the sonar's model to be (3), then (11) should be 0 by Bayesian conditioning because the likelihood of the true value being 7 given an observation of 5 is 0. Regression would reduce the term (11) to one over initial priors:

$$\frac{1}{\gamma} \sum_{x \in \{2,\ldots,11\}} Err(5, x) \times Bel(h = x \wedge h = 7, S_0) \tag{12}$$

where *Err* is the error model from (3). By the condition inside *Bel*, the only valid value for x is 7 for which the prior is .1 but $Err(5, 7)$ is 0. Thus, (11) = (12) = 0. In general, regression is a *recursive* procedure that works iteratively over a sequence of actions discarding one action at a time, and it can be utilized to measure any logical property about the variables, *e.g.* $2\pi \cdot h < 12, h/fuel \leq mileage$, *etc.*

Formally, regression operates at two levels. (Note that this differs slightly from [58, 62].) At the formula level,[9] we introduce an operator $\mathcal{R}$ for regressing formulas, which over equality literals sends the individual terms to an operator $\mathcal{T}$ for regressing terms. These operators proceed by mutual recursion. The fundamental objective of these operators is eliminate *do* symbols. The end result, then, is to transform any expression whose situation term is a successor of S_0, say $do([a_1, a_2], S_0)$, to one about S_0 only, at which point $\mathcal{D}_0$ is all that is needed. As hinted earlier, these operators treat $Bel(\phi, s)$ as though they are special sorts of terms. Throughout the presentation, we assume that the inputs to these operators do not quantify over all situations.

Definition 2. For any term t, we inductively define $\mathcal{T}[t]$:

1. If t is situation-independent (*e.g.* $x, \pi^{2/3}$) then $\mathcal{T}[t] = t$.
2. $\mathcal{T}[g(t_1, \ldots, t_k)] = g(\mathcal{T}[t_1], \ldots, \mathcal{T}[t_k])$,
   where g is any non-fluent function (*e.g.* $\times, +, \mathcal{N}$).
3. For a fluent function f, $\mathcal{T}[f(s)]$ is defined inductively
   (a) if s is of the form $do(a, s')$ then
   $\mathcal{T}[f(s)] = \mathcal{T}[E_f(a)[s']]$
   (b) else $\mathcal{T}[f(s)] = f(s)$

   where, in (a), we use the instance of the RHS of the successor state axiom wrt a, as obtained from (2).
4. $\mathcal{T}[Bel(\phi, s)]$ is defined inductively:
   (a) if s is of the form $do(a, s')$ and a is a noise-free physical action, then
   $$\mathcal{T}[Bel(\phi, s)] = \mathcal{T}[Bel(\psi, s')]$$
   where ψ is $Poss(a, now) \supset \mathcal{R}[\phi[do(a, now)]]$.

[9] For simplicity, in what follows, functional fluents in formulas are only allowed to occur as arguments of an equality literal. It is easy to show that every sentence can be transformed into an equivalent one in the required form, and the transformation is linear in the size of the original sentence, *e.g.* $h \leq 9$ is written as $\exists u\, (h = u \wedge u \leq 9)$.

(b) if s is of the form $do(a, s')$ and a is a sensing action *sense(z)* such that $l(sense(z), s) = Err(z, f_i(s))$ is in $\mathcal{D}$ then

$$\mathcal{T}[Bel(\phi, s)] = \\ \frac{1}{\gamma} \sum_{x_i} Err(z, x_i) \times \mathcal{T}[Bel(\psi, s')]$$

where ψ is $Poss(a, now) \supset \phi \land f_i(now) = x_i$, and γ is the normalization factor and is the same expression as the numerator but ϕ replaced by *true*.

(c) else $\mathcal{T}[Bel(\phi, s)] = Bel(\phi, s)$.

Definition 3. For any formula ϕ, we define $\mathcal{R}[\phi]$ inductively:

1. $\mathcal{R}[t_1 = t_2] = (\mathcal{T}[t_1] = \mathcal{T}[t_2])$
2. $\mathcal{R}[G(t_1, \ldots, t_k)] = G(\mathcal{T}[t_1], \ldots, \mathcal{T}[t_k])$
   where G is any non-fluent predicate (*e.g.* $=, <$).
3. When ψ is a formula, $\mathcal{R}[\neg\psi] = \neg\mathcal{R}[\psi]$,
   $\mathcal{R}[\forall x\psi] = \forall x\mathcal{R}[\psi]$, $\mathcal{R}[\exists x\psi] = \exists x\mathcal{R}[\psi]$.
4. When ψ_1 and ψ_2 are formulas,
   $\mathcal{R}[\psi_1 \land \psi_2] = \mathcal{R}[\psi_1] \land \mathcal{R}[\psi_2]$,
   $\mathcal{R}[\psi_1 \lor \psi_2] = \mathcal{R}[\psi_1] \lor \mathcal{R}[\psi_2]$.
5. $\mathcal{R}[Poss(A(\vec{t}), s)] = \mathcal{R}[\Pi_A(\vec{t}, s)]$,
   where the instance of the RHS of the precondition axiom wrt $A(\vec{t})$ replaces the atom (see Section 2).

This completes the definition of $\mathcal{T}$ and $\mathcal{R}$. We now go over the justifications for the items, starting with the operator $\mathcal{T}$. In item 1, non-fluents simply do not change after actions. In item 2, $\mathcal{T}$ operates over sums and products in a modular manner. In item 3, provided there are remaining *do* symbols, the physics of the domain determines what the conditions must have been in the previous situation for the current value to hold. In item 4, if there is a remainder physical action, part (a) says that belief in ϕ after actions is simply the prior belief about the regression of ϕ, contingent on action executability. Part (b) says that the belief about ϕ after observing z for the true value of f_i is the prior belief for all possible values x_i for f_i that agree with ϕ, times the likelihood of f_i being x_i given z. The appropriateness of parts (a) and (b) depend on the fact that physical actions do not have any sensing aspect, while sensing actions do not change the world. Part (c) simply says that $\mathcal{T}$ stops when no *do* symbols appear in s. We proceed now with the justifications for $\mathcal{R}$. Over equality atoms, $\mathcal{R}$ separates the terms of the equality and sends them to $\mathcal{T}$. Likewise, over non-fluent predicates. Also, $\mathcal{R}$ simplifies over connectives in a straightforward way. When *Poss* is encountered, preconditions take its place.

The main result for $\mathcal{R}$ regarding projection is:

Theorem 4. *Suppose $\mathcal{D}$ is any action theory, ϕ any situation-suppressed formula and δ any action sequence:*

$$\mathcal{D} \models \phi[do(\delta, S_0)] \quad \textit{iff} \quad \mathcal{D}_0 \cup \mathcal{D}_{una} \models \mathcal{R}[\phi[do(\delta, S_0)]]$$

where $\mathcal{D}_{una}$ is the unique name assumption and $\mathcal{R}[\phi[do(\delta, S_0)]]$ mentions only a single situation term, S_0.

Proof: Like in [62], it suffices to show that regression preserves logical equivalence,

$$\mathcal{D} \models \phi[do(\delta, S_0)] \equiv \mathcal{R}[\phi[do(\delta, S_0)]].$$

This is achieved by showing that each step preserves logical equivalence. The process will terminate because each step in regression eliminates the outer *do* from the situation term, and the number of *do* function symbols in any ground situation term is finite. Since each step preserves equivalence, the operator does too as well, resulting in a sentence not containing any actions, so the successor state axioms are no longer required. Thus, $\mathcal{D} \models \mathcal{R}[\phi[do(\delta, S_0)]]$ iff $\mathcal{D}_0 \cup \mathcal{D}_{una} \models \mathcal{R}[\phi[do(\delta, S_0)]]$.

To prove that each step preserves logical equivalence, it suffices to show:

$$\mathcal{D} \models \forall a, s.\ \phi[do(a, s)] \equiv \mathcal{R}[\phi[do(a, s)]].$$

Proof is by induction on the size of ϕ. We treat the size of $\phi[do(a, s)]$ as the size of $\phi[s]$ plus 1. For the base case, Definition 3's items 1 and 2, which then involve Definition 2's items 1, 2, 3(b) and 4(c) are immediate, by means of the standard interpretation of equality and arithmetic symbols. Definition 3's items 3 and 4 follow from the definition of negation and quantifiers, and the hypothesis.

Next, Definition 3's item 5 follows from the precondition axioms, and Definition 2's item 3(a) follows from the successor state axioms. For Definition 4(a), from (B), we have:

$$Bel(\phi, do(a, s)) \doteq \frac{1}{\gamma} \sum_{\{s' : \phi[s']\}} p(s', do(a, s))$$

But, by definition of the successor state axiom of p, $p(s', do(a, s))$ can be obtained from $p(s'', s) \times l(a, s'')$ for $s'' = do(a, s')$ such that $Poss(a, s'') \supset \mathcal{R}[\phi[do(a, s'')]]$. Then, by definition of Bel, we get $L \times Bel(Poss(a, now) \supset \mathcal{R}[\phi[do(a, now)]], s)$. For noise-free actions, we obtain Definition 2's item 4(a) with $L = 1$. For noisy sensing, we note that $l(sense(z), s'')$ is determined by the value of the f fluent at s', and so $L = Err(z, f(s'))$, which is captured in the regression operator by simply summing over possible values of x but testing the actual value at s', akin to an indicator function. ∎

Here, $\mathcal{D}_{una}$ is only needed to simplify action terms [58] *e.g.* from $fwd(4) = fwd(z)$, $\mathcal{D}_{una}$ infers $z = 4$.

The readers may notice many parallels between this regression operator, and the one for categorical knowledge in [62], which should not be surprising as the new operator is a generalization of the previous one.

Now when our goal is to explicitly compute the degrees of belief in the sense of (10), we have the following property for $\mathcal{T}$, which follows as a corollary from the above theorem:

Theorem 5. *Let $\mathcal{D}$ be as above, ϕ any situation-suppressed formula and δ any sequence of actions. Then:*

$$\mathcal{D} \models Bel(\phi, do(\delta, S_0)) = \mathcal{T}[Bel(\phi, do(\delta, S_0))]$$

where $\mathcal{T}[Bel(\phi, do(\delta, S_0))]$ is a term about S_0 only.

Theorem 5 essentially shows how belief about trajectories is computable using beliefs about S_0 only. Note that, since the result of $\mathcal{T}$ is a term about S_0, no sentence outside of $\mathcal{D} - \mathcal{D}_0$ is needed. We now illustrate regression with examples. Using Theorem 5, we reduce beliefs after actions to initial ones. At the final step, standard probabilistic reasoning is applied to obtain the end values.

Example 6. Let $\mathcal{D}$ contain the union of (1), (3) and (5).[10] Then the following equality expressions are entailed by $\mathcal{D}$:

1. $Bel(h = 10 \lor h = 11, S_0) = .2$

   $Bel(h \leq 9, S_0) = .8$

   Terms about S_0 are unaffected by $\mathcal{T}$. So this amounts to inferring probabilities using $\mathcal{D}_0$.

2. $Bel(h = 11, do(fwd(1), S_0))$

   $= \mathcal{T}[\ Bel(h = 11, do(fwd(1), S_0))\ ]$

   $= \mathcal{T}[\ Bel(\ \underline{\mathcal{R}[(h = 11)[do(fwd(1), now)]]}\ , S_0)\ ]$ (i)

   $= \mathcal{T}[Bel(\ \underline{\mathcal{T}[h(do(fwd(1), now))] = \mathcal{T}[11]}\ , S_0)]$ (ii)

[10] Initial beliefs can also be specified for $\mathcal{D}_0$ using Bel, *e.g.* (5) can be replaced in $\mathcal{D}_0$ with $Bel(h = u, S_0) = .1$ for $u \in \{2, \ldots, 11\}$.

$= \underline{\mathcal{T}[Bel(\max(0, h-1) = 11, S_0)]}$ (iii)

$= Bel(\max(0, h-1) = 11, S_0)$ (iv)

$= 0$

First, since action preconditions are all true, *Poss* is ignored everywhere. We underline to emphasize the expressions undergoing transformations. We begin always by applying $\mathcal{T}$ to the main term, in this case getting (i), by means of $\mathcal{T}$'s item 4(a). Next, $\mathcal{R}$'s item 1 is applied in (ii). While $\mathcal{T}[11] = 11$ by $\mathcal{T}$'s item 1, for $\mathcal{T}[h(do(fwd(1), now))]$ we use item 3 and (1) to get:

$$\mathcal{T}[\max(0, h(now) - 1)] = \max(0, h(now) - 1)$$

which is substituted in (ii) to give (iii). Finally, $\mathcal{T}$'s item 4(c) yields (iv), which is a belief term about S_0. Now the only valid value for h in (iv) is 12, but for $h = 12$ the robot has a belief of 0 initially.

3. $Bel(h \leq 5, do(sonar(5), S_0))$

$= \frac{1}{\gamma} \sum_{x \in \{2,\ldots,11\}} Err(5, x) \times \underline{\mathcal{T}[Bel(h = x \wedge h \leq 5, S_0)]}$ (i)

$= \frac{1}{\gamma} \sum_{x \in \{2,\ldots,11\}} Err(5, x) \times Bel(h = x \wedge h \leq 5, S_0)$ (ii)

$$= \frac{1}{\gamma} \Bigg(\frac{1}{3} \cdot Bel(h = 4 \wedge h \leq 5, S_0) + \frac{1}{3} \cdot Bel(h = 5 \wedge h \leq 5, S_0) + \frac{1}{3} \cdot Bel(h = 6 \wedge h \leq 5, S_0) \Bigg) \quad \text{(iii)}$$

$= \frac{1}{\gamma} \left(\frac{1}{3} \cdot Bel(h = 4, S_0) + \frac{1}{3} \cdot Bel(h = 5, S_0) \right)$ (iv)

$= \frac{1}{\gamma} \cdot \frac{2}{30}$

$= 2/3$

where $Err(5, x)$ is the model from (3). First, $\mathcal{T}$'s item 4(b) yields (i), and then item 4(c) yields (ii). Since $Err(5, x)$ is non-zero only for $x \in \{4, 5, 6\}$, (ii) is simplified to (iii) and (iv) resulting in $1/15 \cdot 1/\gamma$. We calculate γ as follows:

$= \sum_{x \in \{2,\ldots,11\}} Err(5, x) \times \mathcal{T}[Bel(h = x \wedge true, S_0)]$ (i′)

$= \sum_{x \in \{2,\ldots,11\}} Err(5, x) \times Bel(h = x, S_0)$ (ii′)

$= 3/30.$

4. Regression for General Domains

We now generalize regression for domains with discrete and continuous variables, for which a *density*-based notion of belief is appropriate. (Physical actions are still noise-free for this section.) The main issue is that when formulating posterior beliefs after sensing, something like Definition 2's item 4(b) will not work. This is because over continuous spaces the belief about any individual point is 0. Therefore, we will be unpacking belief in terms of the density function, *i.e.* in terms of *P*. These $P(\vec{x}, \phi, s)$ terms, which will now also be treated as special sorts of syntactic terms, are separately regressed. (Of course, the regression of weight-based belief can be approached on similar lines.) Recall that $P(\vec{x}, \phi, S_0)$ is simply the *density* of an initial world (where $f_i = x_i$) satisfying ϕ. Formally, term regression $\mathcal{T}$ is defined as follows:

Definition 7. For any term t, we inductively define:

1, 2 and 3 as before.

4. $\mathcal{T}[P(\vec{x}, \phi, s)]$ is defined inductively:

   (a) if s is of the form $do(a, s')$ and a is a physical action then

   $$\mathcal{T}[P(\vec{x}, \phi, s)] = \mathcal{T}[P(\vec{x}, \psi, s')]$$

   where ψ is $Poss(a, now) \supset \mathcal{R}[\phi[do(a, now)]]$.

   (b) if s is of the form $do(a, s')$ and a is a sensing action $sense(z)$ such that $l(sense(z), s) = Err(z, f_i(s))$ is in $\mathcal{D}$, then:

   $$\mathcal{T}[P(\vec{x}, \phi, s)] = Err(z, x_i) \times \mathcal{T}[P(\vec{x}, \psi, s')]$$

   where ψ is $Poss(a, now) \supset \phi \wedge f_i(now) = x_i$.

   (c) else $\mathcal{T}[P(\vec{x}, \phi, s)] = P(\vec{x}, \phi, s)$.

5. $\mathcal{T}[Bel(\phi, s)] = \dfrac{1}{\gamma} \displaystyle\int_{\vec{z}} \mathcal{T}[P(\vec{z}, \phi, s)]$.

$\mathcal{R}$ for formulas is defined as before.

It is worth observing that there is no summation (or integration) symbol when applying $\mathcal{T}$ over noisy sensors because $\mathcal{T}$ over *Bel* expands it first as the integral over the unnormalized density expression $P(\vec{z}, \phi, s)$. In contrast, previously $\mathcal{T}$'s application over a *Bel* term in Definition 2 did not modify the term.

With this new definition, the desired property still holds:

Theorem 8. *Let $\mathcal{D}$ be any action theory, ϕ any situation-suppressed formula and δ any action sequence. Then*

$$\mathcal{D} \models Bel(\phi, do(\delta, S_0)) = \mathcal{T}[Bel(\phi, do(\delta, S_0))]$$

where $\mathcal{T}[Bel(\phi, do(\delta, S_0))]$ is a term about S_0 only.

The proof is analogous to the previous correctness theorem.

Example 9. Consider the following *continuous* variant of the robot example. Imagine a continuous uniform distribution for the true value of h, as provided by (8). Suppose the sonar has the following error profile:

$$l(sonar(z), s) = \text{IF } z \geq 0 \text{ THEN } \mathcal{N}(z - h(s); 0, 4) \text{ ELSE } 0 \tag{13}$$

which says the difference between a nonnegative reading and the true value is normally distributed with mean 0 and variance 4. (A mean of 0 implies there is no systematic bias.) Now, let $\mathcal{D}$ be any action theory that includes (1), (8) and (13). Then the following equalities are entailed by $\mathcal{D}$:

1. $Bel(h = 3 \vee h = 4, S_0) = 0$,

   $Bel(4 \leq h \leq 6, S_0) = .2$

   $\mathcal{T}$ does not change terms about S_0. Here, for example, the second belief term equals $\int_4^6 .1 dx = .2$.

2. $Bel(h \geq 11, do(fwd(1), S_0))$

   $$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[\, P(x, \underline{h \geq 11}, do(fwd(1), S_0)) \,] \tag{i}$$

   $$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[P(x, \underline{\mathcal{R}[\psi]}, S_0)] \tag{ii}$$

where ψ is $(h \geq 11)[do(fwd(1), now)]$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \underline{\mathcal{T}[P(x, \max(0, h-1) \geq 11, S_0)]} \tag{iii}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} P(x, \max(0, h-1) \geq 11, S_0) \tag{iv}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \begin{cases} p(\iota, S_0) & \text{if } \exists \iota.\ h(\iota) = x \wedge h(\iota) \geq 12 \\ 0 & \text{otherwise} \end{cases} \tag{v}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \begin{cases} .1 & \text{if } x \in [2, 12] \text{ and } x \geq 12 \\ 0 & \text{otherwise} \end{cases} \tag{vi}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \begin{cases} .1 & \text{if } x = 12 \\ 0 & \text{otherwise} \end{cases} \tag{vii}$$

$$= 0$$

We use $\mathcal{T}$'s item 5 to get (i), after which item 4(a) is applied. On doing $\mathcal{R}$ in (ii), along the lines of Example 6.2, we obtain (iii). $\mathcal{T}$'s item 4(c) then yields (iv), and stops. In the steps following (iv), we show how P expands in terms of p, and how the space of situations resolves into a mathematical expression, yielding 0.[11]

3. $Bel(h = 0, do(fwd(4), S_0))$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[P(x, \underline{\mathcal{R}[(h = 0)[do(fwd(4), now)]]}, S_0)] \tag{i}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[P(x, \max(0, h-4) = 0, S_0)] \tag{ii}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \begin{cases} .1 & \text{if } x \in [2, 12] \text{ and } x \leq 4 \\ 0 & \text{otherwise} \end{cases} \tag{iii}$$

$$= .2$$

By means of (1), after moving forward by 4 units the belief about h is characterized by a *mixed* distribution because $h = 0$ is accorded a .2 weight (*i.e.* from all points where $h \in [2, 4]$ initially), while $h \in (0, 8]$ are associated with a density of .1. Here, $\mathcal{T}$'s item 5 and 4(a) are triggered, and the removal of $\mathcal{T}$ using 4(c) is not shown. The end result is that the density function is integrated for $2 \leq x \leq 4$ leading to .2. (γ is 1.)

4. $Bel(h = 4, do(fwd(-4), do(fwd(4), S_0)))$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[P(x, \exists u.\ h = u \wedge \\ 4 = \max(0, u + 4), do(fwd(4), S_0))] \tag{i}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[P(x, \exists u.\ u = \max(0, h-4) \wedge \\ 4 = \max(0, u + 4), S_0)] \tag{ii}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \begin{cases} .1 & \text{if } x \in [2, 12],\ x \leq 4 \\ 0 & \text{otherwise} \end{cases} \tag{iii}$$

$$= .2$$

We noted above that the point $h = 4$ gets a .2 weight on executing $fwd(4)$, after which it obtains a h value of 0. The weight is *retained* on reversing by 4 units, with the point now obtaining a h value of 4. The derivation invokes two applications of $\mathcal{T}$'s item 4(a). We skip the intermediate $\mathcal{R}$ steps. (γ evaluates to 1.)

[11] Given certain assumptions, it is possible to further reduce logical expressions involving fluents to a mathematical expression using only those variables that appear in the integral. We expand on this in a longer version of the paper.

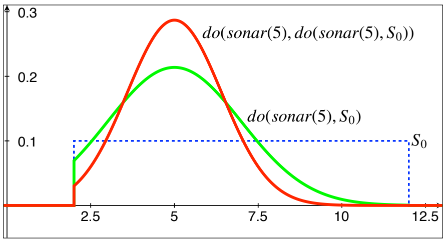

Figure 3: Belief density change for h at S_0 (in blue), after sensing 5 (in green) and after sensing 5 twice (in red).

5. $Bel(h = 4, do(fwd(4), do(fwd(-4), S_0)))$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[P(x, \exists u.\ u = \max(0, h + 4) \ \wedge \ 4 = \max(0, u - 4), S_0)] \quad \text{(i)}$$

$= 0$

Had the robot moved away first, no "collapsing" of points takes place, h remains a continuous distribution and no point is accorded a non-zero weight. $\mathcal{T}$ steps are skipped but they are symmetric to the one above, *e.g.* compare (i) here and (ii) above. But then the density function is non-zero only for the individual $h = 4$.

6. $Bel(4 \leq h \leq 6, do(sonar(5), S_0))$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{N}(5 - x; 0, 4) \times \mathcal{T}[P(x, \psi, S_0)] \quad \text{(i)}$$

where ψ is $h = x \wedge 4 \leq h \leq 6$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \begin{cases} .1 \cdot \mathcal{N}(5 - x; 0, 4) & \text{if } x \in [2, 12],\ x \in [4, 6] \\ 0 & \text{otherwise} \end{cases}$$

$\approx .41$

We obtain (i) after $\mathcal{T}$'s item 5 and then 4(b) for sensing actions. That is, belief about $h \in [4, 6]$ is sharpened after observing 5. Basically, we are integrating a function that is 0 everywhere except when $4 \leq x \leq 6$ where it is $.1 \times \mathcal{N}(5 - x; 0, 4)$, normalized over $2 \leq x \leq 12$.

7. $Bel(4 \leq h \leq 6, do(sonar(5), do(sonar(5), S_0))$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{N}(5 - x; 0, 4) \times \underline{\mathcal{T}[P(x, \psi, s))]} \quad \text{(i)}$$

where $s = do(sonar(5), S_0)$, ψ is $h = x \wedge 4 \leq h \leq 6$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} [\mathcal{N}(5 - x; 0, 4)]^2 \times \mathcal{T}[P(x, \psi, S_0)] \quad \text{(ii)}$$

$\approx .52$

As expected, two successive observations of 5 sharpens belief further. Derivations (i) and (ii) follow from $\mathcal{T}$'s item 5, and two successive applications of item 4(b). Thus, we are to integrate $.1 \times [\mathcal{N}(5 - x; 0, 4)]^2$ between $[4, 6]$ and normalize over $[2, 12]$. These changing densities are plotted in Figure 3.

5. Two Special Cases

Regression is a general property for computing properties about posteriors in terms of priors after actions. It is therefore possible to explore limited cases, which might be appropriate for some applications. We present two such cases.

Conjugate distributions

Certain types of systems, such as Gaussian processes, admit an effective propagation model [69]. The same advantages can be observed in our framework. We illustrate this using an example. Assume a fluent f, and suppose $\mathcal{D}_0$ is the union of (P3), (P1) and the following specification:

$$p(\iota, S_0) = \mathcal{N}(f(\iota); \mu_1, {\sigma_1}^2)$$

which stipulates that the true value of f is believed to be normally distributed. Assume the following sensor in $\mathcal{D}$:

$$l(\mathit{sense}(z), s) = \mathcal{N}(z - f(s); \mu_2, {\sigma_2}^2)$$

Then it is easy to show that estimating posteriors yields a product of Gaussian density function (that is also a Gaussian density function [16]), which is inferred by $\mathcal{T}$:

$$\begin{aligned} &\mathcal{T}[\mathit{Bel}(b \leq f \leq c, \mathit{do}(\mathit{sense}(z), S_0))] = \\ &\qquad \frac{1}{\gamma} \int_b^c \mathcal{N}(x; \mu_1, {\sigma_1}^2) \cdot \mathcal{N}(z - x; \mu_2, {\sigma_2}^2) \mathrm{d}x \end{aligned}$$

Distribution transformations

Certain actions affect priors in a characteristically simple manner, and regression would account for these changes as an appropriate function of the initial belief state. We illustrate two instances using Example 9. First, consider an action $\mathit{grasp}(z)$ that grabs object z. Because the action of grasping does not affect h by way of (1), we get:

$$\mathcal{T}[\mathit{Bel}(h \leq b, \mathit{do}(\mathit{grasp}(\mathit{obj5}), S_0))] = \mathit{Bel}(h \leq b, S_0)$$

So no changes to h's density are required. Second, consider ground actions with the property that two distinct values of f do not become the same after that action, *e.g.*, for initial states this means:

$$\forall \iota, \iota'.\ f(\iota) \neq f(\iota') \supset f(\mathit{do}(a, \iota)) \neq f(\mathit{do}(a, \iota'))$$

Think of $\mathit{fwd}(-4)$ that agrees with this, but $\mathit{fwd}(4)$ need not. We can show that such actions "shift" priors:

$$\mathcal{T}[\mathit{Bel}(h \leq b, \mathit{do}(\mathit{fwd}(-n), S_0))] = \mathit{Bel}(h \leq b - n, S_0)$$

Intuitively, the probability of h being in the interval $[b, c]$, irrespective of the distribution family, is the same as the probability of $h \in [b + n, c + n]$ after $\mathit{fwd}(-n)$. Thus, regression derives the initial interval given the current one.

6. Regression over Noisy Actions

The regression operator thus far was limited to noise-free actions and noisy sensing. We first show how the logical account is extended to handle noisy actions (following [10]). We then extend the regression operator.

The idea behind noisy actions is that an agent might attempt a physical move of 3 units, but as a result of the limited accuracy of effectors, actually move (say) 3.094 units. Thus, unlike sensors, where the reading is nondeterministic, *observable,* but does not affect fluents, the outcome of noisy actions is nondeterministic, *unobservable* and changes fluent properties. Of course, when attempting to move 3 units, the agent knows that an actual move by 3.094 units is much more likely than 30.94 units, and that is reflected in a noise model.

While [2] relied on Golog [43] to capture noisy actions, a simpler proposal is given in [10]. Here, instead of an action like $\mathit{fwd}(x)$, we use an action $\mathit{fwd}(x, y)$ where x is the intended motion (3 units) known to the agent and y is the actual motion (say, 3.094 units) unknown to the agent. The successor state axiom for h would be written to reflect the fact that its value is changed by the second argument:

$$\begin{aligned} h(\mathit{do}(a, s)) = \ &\text{IF } \exists x, y (a = \mathit{fwd}(x, y)) \\ &\quad \text{THEN } \max(0, h(s) - y) \text{ ELSE } h(s). \end{aligned} \tag{14}$$

Precondition axioms would stipulate conditions for doing the intended action, as usual. For example,

$$Poss(fwd(x, y), s) \equiv x > 0$$

says that as long as the intended value is not zero units, all possible values for y are permitted. Unlike noise-free actions where the likelihood is always 1, noisy actions would have non-trivial l-axioms and might further constrain unintended outcomes:

$$l(fwd(x, y), s) = \text{IF } y = x \text{ THEN } .9 \text{ ELSE } (\text{IF } y = 0 \text{ THEN } .1 \text{ ELSE } 0). \tag{15}$$

Together with the precondition axiom that disallows zero and negative values for x, the l-axiom says that the likelihood of y being exactly x is .9, that the likelihood of no move happening is .1 and all other possibilities are improbable. Analogously, a Gaussian noise model might be defined as follows:

$$l(fwd(x, y), s) = \mathcal{N}(y; x, 1).$$

This axiom says that the actual value moved is normally distributed around the intended value, with a variance of 1.

In general, we assume that for each noisy action $act(x, y)$, the action theory $\mathcal{D}$ includes an axiom of the form:[12]

$$l(act(x, y), s) = Err_{act}(x, y).$$

Of course, we will also need to adapt the definition of *Bel* to account for not knowing the actual amount moved, by integrating over the possible choices. The intuition is this: any of these outcomes are considered possible, so the belief state of the agent would entertain all of these outcomes as possible successor situations.

Formally, recall that $Bel(\phi, s)$ was previously defined in terms of $P(\vec{x}, \phi, do(\delta, s))$ wrt some ground action sequence δ of the form $[a_1(c_1), \ldots, a_k(c_k)]$. In the noisy action setting, we are dealing with ground action sequences of the form $\delta = [a_1(c_1, d_1), \ldots, a_k(c_k, d_k)]$. So first define:

$$\begin{array}{l} P(\vec{x} \cdot \vec{v}, \phi, do(\delta, S_0)) \doteq \\ \qquad \text{IF } \exists \iota.\ \bigwedge f_i(\iota) = x_i \land \phi[do(\delta', \iota)] \\ \qquad \qquad \text{THEN } p(do(\delta', \iota), do(\delta, S_0)) \\ \qquad \text{ELSE } 0 \end{array}$$

where δ' is a possible outcome $[a_1(c_1, v_1), \ldots, a_k(c_k, v_k)]$, which has free variables $v_1, \ldots, v_k$ that are to be ground wrt the domain of the integration operator in *Bel*. Basically, for each possible instantiation of $\vec{v}$, a different successor situation $do(\delta', \iota)$ is realized. So, the above expression checks whether ϕ holds at this successor situation, and then retrieves the p-value of that situation relative to the real world $do(\delta, S_0)$. It then follows that $Bel(\phi, s)$ is defined as:

$$Bel(\phi, s) \doteq \frac{1}{\gamma} \int_{\vec{x}} \int_{\vec{v}} P(\vec{x} \cdot \vec{v}, \phi, s) \tag{16}$$

We are now ready to define term and formula regression:

Definition 10. We define $\mathcal{T}$ and $\mathcal{R}$ as in Definition 7 except with the following change to item 4(b):

If s is of the form $do(a, s')$ and a is a sensing action $sense(z)$ such that $l(sense(z), s) = Err(z, f_i(s))$ is in $\mathcal{D}$, then:

$\mathcal{T}[P(\vec{x}, \phi, s)] = Err(z, x_i) \times \mathcal{T}[P(\vec{x}, \psi, s')]$

where ψ is $Poss(a, now) \supset \phi \land f_i(now) = x_i$.

If s is of the form $do(a, s')$ and a is a noisy action $act(n, m)$ such that $l(act(u, v), s) = Err(u, v)$ is in $\mathcal{D}$ then:

$\mathcal{T}[P(\vec{x}, \phi, s)] = \int_v Err(n, v) \times \mathcal{T}[P(\vec{x} \cdot v, \psi(v), s')]$

where $\psi(v)$ is $Poss(act(n, v), now) \supset \mathcal{R}[\phi[do(act(n, v), now)]]$.

[12] For ease of presentation, we are assuming that noisy actions only come with two arguments, one standing for the intended value and the other for the actual outcome. It is straightforward to generalize the likelihood axiom for k-ary actions, or even handle context-dependent noisy actions like we have enabled for noisy sensing. This is possible by allowing fluents as additional arguments in Err_{act}.

So, while $\mathcal{T}$ for noisy sensing is the same as before, when we have a ground noisy action $act(n,m)$, we observe the following. First, since the agent does not actually observe m, a fresh variable v is introduced, which is to be the integration variable, and the density is multiplied by the noise model. The density expression then regresses the formula wrt $act(n,v)$. In fact, as we show below, this regression is best realized by keeping v as a free variable, because that then gives us a single expression over which we can integrate.

Most significantly, the desired property still holds, with an analogous proof:

Theorem 11. *Let $\mathcal{D}$ be any action theory, ϕ any situation-suppressed formula and δ any action sequence. Then*

$$\mathcal{D} \models Bel(\phi, do(\delta, S_0)) = \mathcal{T}[Bel(\phi, do(\delta, S_0))]$$

where $\mathcal{T}[Bel(\phi, do(\delta, S_0))]$ is a term about S_0 only.

Example 12. Let us consider the same domain as in Example 9, but in using the noisy action model from (14) and (15). Then we get the following entailments from $\mathcal{D}$:

1. $Bel(h = 3 \vee h = 4, S_0) = 0$,

   $Bel(4 \leq h \leq 6, S_0) = .2$

   $\mathcal{T}$ does not change terms about S_0. Here, for example, the second belief term equals $\int_4^6 .1 \mathrm{d}x = .2$.

2. $Bel(h \geq 11, do(fwd(1,0), S_0))$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \mathcal{T}[\, P(x, \underline{h \geq 11}, do(fwd(1,0), S_0))\,] \tag{i}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \int_{v \in \mathbb{R}} Err(1,v) \times \mathcal{T}[P(x \cdot v, \underline{\mathcal{R}[\psi(v)]}, S_0)] \tag{ii}$$

where $\psi(v)$ is $(h \geq 11)[do(fwd(1,v), now)]$ and $Err(1,v)$ is the error function from (15)

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \int_{v \in \mathbb{R}} Err(1,v) \times \underline{\mathcal{T}[P(x \cdot v, \max(0, h - v) \geq 11, S_0)]} \tag{iii}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \int_{v \in \mathbb{R}} Err(1,v) \times P(x \cdot v, \max(0, h - v) \geq 11, S_0) \tag{iv}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \int_{v \in \mathbb{R}} \begin{cases} p(\iota, S_0) \times Err(1,v) & \text{if } \exists \iota.\ h(\iota) = x \wedge h(\iota) \geq 11 + v \\ 0 & \text{otherwise} \end{cases} \tag{v}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \int_{v \in \mathbb{R}} \begin{cases} .1 \times Err(1,v) & \text{if } x \in [2,12] \text{ and } x \geq 11 + v \\ 0 & \text{otherwise} \end{cases} \tag{vi}$$

$$= \frac{1}{\gamma} \int_{x \in \mathbb{R}} \int_{v \in \mathbb{R}} \begin{cases} .1 \times .9 & \text{if } x \in [2,12] \text{ and } x \geq 12 \\ .1 \times .1 & \text{if } x \in [2,12] \text{ and } x \geq 11 \\ 0 & \text{otherwise} \end{cases} \tag{vii}$$

$$= 0.01$$

The application of $\mathcal{T}$ and $\mathcal{R}$ proceeds in the same fashion as in Example 9, with the following notable changes. In (ii), $\mathcal{T}$ for noisy actions is applied, which introduces a new integration symbol (over v), and the formula to be regressed has v as a free variable. Moreover, the noise model is multiplied to the current density. As the successive steps show, regressing the formula over actions by using the RHS of successor state axioms continues to keep v as a free variable, until (vi). The noise model is such that v takes the value 1 with a probability of .9, and a value 0 with a probability of .1. We know from Example 9 that when the move happens by 1 unit, the regressed formula is essentially $h \geq 12$, which has a probability of 0. So the only non-zero event is when no move happens, in which case we are interested in the prior belief of $h \geq 11$, which is .1, but further multiplied by the likelihood of that outcome, which is also .1, and so we obtain 0.01.

7. Progression

In the worst case, regressed formulas are exponentially long in the length of the action sequence [58], and so it has been argued that for long-lived agents like robots, continually updating the current view of the state of the world, is perhaps better suited. Lin and Reiter [47] proposed a theory of progression for the classical situation calculus. What we are after is an account of progression for probabilistic beliefs in the presence of stochastic noise. However, subtleties arise with the p fluent even in simple continuous domains. For example, consider (1), where the noise-free action *fwd*(z) that moves the robot towards the wall but stopping when the robot hits the wall:

$$h(do(a,s)) = \text{IF } \exists z(a = fwd(z)) \text{ THEN } \max(0, h(s) - z) \text{ ELSE } h(s).$$

If the robot were to begin with (6) and perform the action *fwd*(4), beliefs about the new value of h become much more complex. Roughly, those points where $h \in [2,4]$ initially are mapped to a single point $h = 0$ that should then obtain a probability *mass* of .2, while the other points retain their initial *density* of .1. In effect, a probability density on h is transformed into a mixed density / distribution (and the (P3) assumption no longer holds). In the previous sections, we dealt with this issue using *regression*: beliefs are regressed to the initial state where (P3) does hold, and all the actual belief calculations can be done in the initial state.

In this section, we develop a logical theory of progression for basic action theories where such mixed distributions do not arise.[13] We provide a simple definition to support this, and then discuss general syntactic restrictions that satisfy this requirement.

7.1. Invertible Action Theories

Our formulation of progression rests on introducing a new class of basic action theories, called *invertible action theories*. Recall that successor state axioms are of the general form: $f(do(a,s)) = E_f(a)[s]$, which tells us how the value of f changes from s to $do(a,s)$. We will now base our work on the following question: given the value of f at $do(a,s)$, what is the value of f at s?

Definition 13. Given a basic action theory $\mathcal{D}$, a fluent f is said to be *invertible* if there is an expression $H_f(a)$ uniform in *now* such that $\mathcal{D} \models f(s) = H_f(a)[do(a,s)]$. We say that $\mathcal{D}$ is invertible if every fluent in the theory is invertible.

Intuitively, a fluent is invertible when we can find a dual formulation of its successor state axiom, that is, where we can characterize the *predecessor* value of a fluent in terms of its current value.

There are three syntactic conditions on a basic action theory $\mathcal{D}$ that are sufficient to guarantee its invertibility:

i. There is an ordering on fluents such that all the fluents that appear in $E_f(a)$ other than f are earlier in the ordering.

ii. Any situation term in $E_f(a)$ appears as an argument to one of the fluents.

iii. The mapping from the value of $f(s)$ to the value of $f(do(a,s))$ given by $E_f(a)$ is bijective [70]. (This is understood in the usual set-theoretic sense.)

[13] Its worth remarking that there is nothing inherently problematic about mixed distributions as far as the definability of progressed database is concerned. Indeed, for the above example, contrast the initial theory with the one after *fwd*(4) below:

$$p(\iota, s) = \begin{cases} .1 & \text{if } h(\iota) \in [2,12] \\ 0 & \text{otherwise} \end{cases} \quad \text{versus} \quad p(\iota, s) = \begin{cases} .2 & \text{if } h(\iota) = 0 \\ .1 & \text{if } h(\iota) \in (0,8] \\ 0 & \text{otherwise} \end{cases}$$

Both of these are well-defined p-specifications. Be that as it may, it is not immediately obvious what the account of progression should look like, in terms of the general syntactic rules that allow us to update the database so as to yield the latter initial theory. There is also an issue with the definition of *Bel*, because we now need to sum over the values of h that are discrete, and integrate over the rest. Although this can be handled (see [10]), it makes the account slightly more involved. Our proposal eschews these complications in a simple yet reasonable manner, and moreover, as we shall shortly see, subsumes some of the analytical cases seen in the literature.

Before considering some examples, here is the result:

Theorem 14. *If a basic action theory satisfies (i), (ii) and (iii) above, then it is invertible.*

Proof: The proof is by induction on the ordering given by (i). By (iii), we can take $f(do(a, s)) = E_f(a)[s]$ and solve for $f(s)$, obtaining an equation $f(s) = H$, where H mentions $f(do(a, s))$ and possibly other fluents $f'(s)$ that appear earlier in the ordering. By induction, each $f'(s)$ in H can be replaced by $H_{f'}(a)[do(a, s)]$. By (ii), the result will then be uniform in $do(a, s)$, and thus we obtain a formula $H_f(a)$ where $\mathcal{D} \models f(s) = H_f(a)[do(a, s)]$ as desired. ∎

Example 15. Let us consider the setting from Figure 2. In particular, the effect on v might be described by:

$$v(do(a, s)) = u \equiv \exists z(a = mv(z) \wedge u = v(s) + z) \vee \\ u = v(s) \wedge \neg\exists z(a = mv(z)).$$

This says that $mv(z)$ is the only action affecting v, thereby incorporating a solution to the frame problem [58]. We would now equivalently write this as:

$$v(do(a, s)) = \text{IF } \exists z(a = mv(z)) \\ \text{THEN } v(s) + z \text{ ELSE } v(s). \tag{17}$$

Consider (17). This trivially satisfies (i) and (ii). The mapping from $v(s)$ to $v(do(a, s))$ is bijective and so (iii) is satisfied also. (In general, any $E_f(a)$ that is restricted to addition or multiplication by constants will satisfy (iii).) So the fluent is invertible and we have $v(s) = H_v(a)[do(a, s)]$, where $H_v(a)$ is IF $\exists z(a = mv(z))$ THEN $v - z$ ELSE v.

Example 16. Consider (1). Here the mapping is not bijective because of the *max* function and the fluent h is not invertible. If $h(do(\alpha, s)) = 0$ where $\alpha = fwd(4)$, then the value of $h(s)$ cannot be determined and can be anything less than 4.

Example 17. Consider a successor state axiom like this:

$$v(do(a, s)) = \text{IF } \exists z(a = mv(z)) \text{ THEN } (v(s))^z \text{ ELSE } v(s).$$

For $\alpha = mv(2)$, we obtain a squaring function, which is not bijective. Indeed, from $v(do(\alpha, s)) = 9$, one cannot determine whether $v(s)$ was -3 or 3, and the fluent is not invertible.

Example 18. Consider this successor state axiom for compound interest, where v denotes the accumulated value, *rate* denotes the annual interest rate, and *lapse(z)* denotes the number of years the interest was allowed to accumulate:

$$v(do(a, s)) = \text{IF } \exists z(a = lapse(z)) \wedge relief(s) = 0 \\ \text{THEN } v(s) \cdot (1 + rate(s))^z \text{ ELSE } v(s).$$

Suppose further:

$$rate(do(a, s)) = \text{IF } \exists z(a = change(z)) \text{ THEN } z + rate(s) \text{ ELSE } rate(s).$$

$$relief(do(a, s)) = \text{IF } a = toggle \text{ THEN } 1 - relief(s) \text{ ELSE } relief(s).$$

Here, the interest rate influences the accumulated value over the lapsed time, and *relief* being true stops the accumulation of interest. This theory is invertible, and $H_v(a)$ is given by

$$H_v(a) = \text{IF } \exists z(a = lapse(z)) \wedge relief = 0 \\ \text{THEN } v/(1 + rate)^z \text{ ELSE } v.$$

That is, because $lapse(z)$ does not affect *relief* and *rate*, we simply invert the successor state axiom for v and relativize everything to $do(a, s)$. If (say) the action *lapse(z)* also affected *rate*, by the ordering in (i), we would first obtain the H-expression for *rate* and use it in the H-expression for v.

Note that the bijection property does not prevent us from using non-bijective functions, such as squares, in the successor state axiom of v, provided that these only apply to the other fluents. (The remaining fluents essentially behave as constants at any given situation.) In our experience, many commonly occurring successor state axioms are invertible.

Before concluding our development of invertible theories, let us reflect on the sufficiency conditions (i), (ii) and (iii). It should be clear that (iii), in fact, is also a necessary condition.

Theorem 19. *If a fluent is invertible then (iii) must hold.*

Proof: Suppose f is invertible but (iii) does not hold. This means that the mapping from the value of $f(s)$ to the value of $f(do(a,s))$ given by $E_f(a)$ is not bijective. However, a function is invertible iff it is a bijection [13]. In our context, this implies that the value of $f(s)$ cannot be uniquely determined from the value of $f(do(a,s))$. Thus, there cannot be an expression $H_f(a)$ such that $\mathcal{D} \models f(s) = H_f(a)[do(a,s)]$. Contradiction. ∎

What about the necessity of (i) and (ii)? The main reason we insisted on these conditions in the first place is because obtaining the value of $f(s)$ from $f(do(a,s))$ becomes straightforward by using the fluent values at the start of the order. To see why dropping these conditions makes the setting challenging, suppose f and g are the only two fluents in a basic action theory, *act* is the only action, and suppose we have the following successor state axioms:

$$\begin{aligned} f(do(a,s)) = {} & \text{IF } (a = act) \\ & \quad \text{THEN } expr_1(f(s), g(s)) \text{ ELSE } f(s). \\ g(do(a,s)) = {} & \text{IF } (a = act) \\ & \quad \text{THEN } expr_2(f(s), g(s)) \text{ ELSE } g(s). \end{aligned}$$

Here, $expr_i$ could denote sums or products, or any other 2-ary bijective function. Suppose we are now given the values of $f(do(a,s))$ and $g(do(a,s))$, and as motivated earlier, we are to recover the values of $f(s)$ and $g(s)$. To obtain the value of $f(s)$, then, we would need the value of $f(do(a,s))$, which is given, but also the value of $g(s)$, which has to be obtained. To obtain the value of $g(s)$, we would need the value of $g(do(a,s))$, which is given, but also the value of $f(s)$. In other words, we are given a system of equations:

$$\begin{aligned} x' &= expr_1(x, y) \\ y' &= expr_2(x, y) \end{aligned}$$

where the values of $\{x', y'\}$ are known and denote the values of $\{f(do(a,s)), g(do(a,s))\}$ respectively, and we are to solve for $\{x, y\}$ that denote the values of $\{f(s), g(s)\}$ respectively. In many cases, such systems can, of course, be solved; for example, if $x' = x + y$ and $y' = y - x$, then $x = x' - (y' + x) = x' - y' - x$, which means $2x = x' - y'$. Once we obtain the value of x, we can obtain the value of y analogously. In general, however, solving such a system of equations may not always be possible, at least in an exact manner.

Thus, (i) and (ii) are not necessary in order to obtain H-expressions, but simplify the treatment considerably since we only need to invert the function expressed in $E_f(a)$, and obtain the values of the fluents according to the order. It would be interesting to see whether for the basic action theories considered in the literature, even if (i) and (ii) do not hold, $H_f(a)$ can be obtained in an exact manner making (iii) both sufficient and necessary for this class of theories.

7.2. Classical Progression

We now are prepared for a definition of progression that applies to any invertible basic action theory. Note that the definition of invertibility imposes no constraint on $\mathcal{D}_0$. So the definition in this section is general in that the p may appear in $\mathcal{D}_0$ in an unrestricted way, such as in (7). Given such a theory $\mathcal{D}_0 \cup \Sigma$ and a ground action α, we define a transformation $\mathcal{D}_0'$ such that $\mathcal{D}_0' \cup \Sigma$ agrees with $\mathcal{D}_0 \cup \Sigma$ on the future of α. Then, in the next section, we will consider how $\mathcal{D}_0'$ grows as a result of this progression.

To start with, let us first consider the simpler case of progression for a $\mathcal{D}_0$ that does not mention the p fluent (and the quantification over initial situations that comes with it), and so where $\mathcal{D}_0$ is uniform in S_0. In this case, because we are assuming a finite set of nullary fluents, any basic action theory can be shown to be *local-effect* [48], where progression is first-order definable. The new theory is computed by appealing to the notion of *forgetting* [46]. If the basic action theory is invertible, however, the progression can *also* be defined in another way. Let $\mathcal{D}_0'$ be $\mathcal{D}_0$ but with any $f(S_0)$ term in it replaced by $H_f(\alpha)[S_0]$.

Theorem 20. *Let $\mathcal{D}_0 \cup \Sigma$ be any invertible basic action theory not mentioning p and α any ground action. Then for any $\mathcal{L}$-formula ϕ rooted in now*

$$\mathcal{D}_0 \cup \Sigma \models \phi[do(\alpha, S_0)] \ \textit{ iff } \ \mathcal{D}_0' \cup \Sigma \models \phi[S_0].$$

The proof for this result is the first part of the proof for the below theorem.[14]

Example 21. Consider (17), and the $H_v(a)$ from Example 15. Suppose $\mathcal{D}_0 = \{v(S_0) > 10\}$. Then:

$$\begin{aligned} \mathcal{D}_0' &= (H_v(mv(3))[S_0]) > 10 \\ &= (\text{IF } \exists z(mv(z) = mv(3)) \\ &\qquad \text{THEN } v(S_0) - z \text{ ELSE } v(S_0)) > 10 \\ &= (v(S_0) - 3 > 10) \\ &= (v(S_0) > 13). \end{aligned}$$

Therefore, as expected, the progression of $v(S_0) > 10$ wrt a noise-free motion of 3 units is $v(S_0) > 13$. (The unique name axiom and arithmetic are used in the simplification.)

7.3. Progressing Degrees of Belief

There are two main complications when progressing beliefs wrt noisy sensors and actions. First, the p fluent will have to take the likelihood of the action α into account. Second, $\mathcal{D}_0$ need not be uniform in S_0, since p typically requires quantification over initial situations (as in (6), for example). This leads to the following definition:

Definition 22. Let $\mathcal{D}_0 \cup \Sigma$ be an invertible basic action theory and α be a ground action of the form $A(\vec{t})$ where $\vec{t}$ is uniform in *now*.[15] Then $Pro(\mathcal{D}_0, \alpha)$ is defined as $\mathcal{D}_0$ with the following substitutions:

- $p(\iota, S_0)$ is replaced by $\dfrac{p(\iota, S_0)}{Err_A(\vec{t})[\iota]}$;
- every other fluent term $f(u)$ is replaced by $H_f(\alpha)[u]$.

Here, $Err_A(\vec{x})$ refers to the RHSof the likelihood axiom for $A(\vec{x})$.

The main result of this paper is the correctness of this definition of progression:

Theorem 23. *Under the conditions of the definition above, let $\mathcal{D}_0' = Pro(\mathcal{D}_0, \alpha)$. Suppose that $\mathcal{D}_0 \models (Err_A(\vec{t}) \neq 0)[S_0]$. Then for any $\mathcal{L}$-formula ϕ rooted in now,*

$$\mathcal{D}_0 \cup \Sigma \models \phi[do(\alpha, S_0)] \ \textit{ iff } \ \mathcal{D}_0' \cup \Sigma \models \phi[S_0].$$

Proof: Following [47], to show that $\mathcal{D}_0' \cup \Sigma$ is the progression of $\mathcal{D}_0 \cup \Sigma$, it suffices to show that for any model M, M is a model of $\mathcal{D}_0' \cup \Sigma$ iff there is a model M' of $\mathcal{D}_0 \cup \Sigma$ such that for any ϕ rooted in *now*, $M \models \phi[S_0]$ iff $M' \models \phi[do(\alpha, S_0)]$.

For any M, let M' be exactly like M except for the interpretation of $f_1, \ldots, f_k, p$ initially, which is specified separately below. That is, M and M' have the same domains for actions and objects, interpret situation-independent symbols exactly, and also interpret $do, Poss$ exactly. It then follows that it suffices to show that for all fluents f (including p), $M \models f(S_0) = n$ iff $M' \models f(do(\alpha, S_0)) = n$. If that were true, by the successor state axioms and Σ, the more general case of formulas ϕ rooted in *now* would follow by an induction argument.

[14] We deviate from the formulation in [47] in one minor way: the classical account defines the progression of $\mathcal{D}_0$ to be a set of sentences that are uniform in $do(\alpha, S_0)$, and require that $\mathcal{D}_0$ and $\mathcal{D}_0'$ agree on all formulas about $do(\alpha, S_0)$ and its future situations. (Note that ϕ in the theorem is permitted to be rooted in *now*, which means it can capture future situations by being of the form $\psi[do([\alpha_1, \ldots, \alpha_n], now)]$.) So, the theorem would instead be stated as saying that $\mathcal{D}_0 \cup \Sigma \models \phi[do(\alpha, S_0)]$ iff $\mathcal{D}_0' \cup \Sigma \models \phi[do(\alpha, S_0)]$. We are choosing to instead formulate $\mathcal{D}_0'$ as being uniform in S_0, as we think its somewhat simpler to read when we invert the successor state axioms.

The original variant is easily obtained by simply replacing all occurrences of S_0 in $\mathcal{D}_0'$ by $do(\alpha, S_0)$. Replacing the situation terms in a progressed theory is not uncommon; see, for example, the lifting of predicate symbols in [48].

[15] In the most common case (like noise-free or sensing actions), the arguments to the action would simply be a vector of constants.

The proof is fairly easy to establish because by construction alone, we have a precise space of initial situations via (P3), and the bijective mapping from the value of $f(s)$ to the value of $f(do(\alpha, s))$ via invertible theories. We first treat the regular fluents $f_1, \ldots, f_k$ and treat p soon after. Suppose the values of the regular fluents at S_0^M is $\langle n_1, \ldots, n_k\rangle \in \mathbb{R}^k$. Then let the value for the fluent f_i at $S_0^{M'}$ be the *inverted value*, which is given as: $M' \models f_i(S_0) = (H_{f_i}(\alpha)[do(\alpha, S_0)])\ (f_1(do(\alpha, S_0)), \ldots, f_k(do(\alpha, S_0))/n_1, \ldots, n_k)$, that is, the fluent terms in $(H_{f_i}(\alpha)[do(\alpha, S_0)])$ are replaced by $n_1, \ldots, n_k$, which is supposed to be the the values of fluents at $do(\alpha, S_0)$ as per S_0^M. Because of invertibility, it immediately follows that for every f_i, $M \models f_i(S_0) = n$ iff $M' \models f_i(do(\alpha, S_0)) = n$.

The case for p remains. Recall that M and M' agree on the space of initial situations. Following the construction for S_0 above, it should be clear that for any initial situation s^M, there is an initial situation $s^{M'}$ such that for every f_i, $M \models f(s) = n$ iff $M' \models f(do(\alpha, s)) = n$. We now show that the p-value of s^M is exactly the same as the p-value of $s^{M'}$. It then follows that for any situation suppressed formula ϕ, $M \models Bel(\phi, S_0) = n$ iff $M' \models Bel(\phi, do(\alpha, S_0)) = n$.

To show the mapping of p values, consider that the general form is that $p(\iota, S_0) = \theta(\iota)$ is in $\mathcal{D}_0$, and by construction, $\mathcal{D}'_0$ includes $p(\iota, S_0) = \theta[\iota] \times Err_A(\vec{t})[\iota]$. Then for the initial situation s^M, its p-value is given as: $M \models p(s, S_0) = (\theta \times Err_A(\vec{t}))\ (f_1(s), \ldots, f_k(s)/n_1, \ldots, n_k)$, that is, the expressions have the the fluents replaced by their values $\langle n_1, \ldots, n_k\rangle \in \mathbb{R}^k$ at s^M. Now $s^{M'}$ obtains a p-value: $M' \models p(s, S_0) = \theta\ (f_1(s), \ldots, f_k(s)/n'_1, \ldots, n'_k)$, where $\langle n'_1, \ldots, n'_k\rangle \in \mathbb{R}^k$ are the values of the fluents at $s^{M'}$. Clearly by the successor state axiom for p, the p-value of $do^{M'}(\alpha^{M'}, s^{M'})$ is the same as the p-value of s^M. ∎

This theorem gives us the desired property for *Bel* (which is defined in terms of p) as a corollary:

Corollary 24. *Suppose $\mathcal{D}_0, \Sigma, \mathcal{D}'_0, \phi$, and α are as above. Then for all real numbers n:*

$$\begin{aligned} &\mathcal{D}_0 \cup \Sigma \models Bel(\phi, do(\alpha, S_0)) = n \\ &\quad \textit{iff} \quad \mathcal{D}'_0 \cup \Sigma \models Bel(\phi, S_0) = n. \end{aligned}$$

Thus the degree of belief in ϕ after a physical or sensing action is equal to the initial belief in ϕ in the progressed theory.

We now present some examples, considering, in turn, noise-free actions, noisy sensing and finally noisy actions.

Example 25. Let us consider an action theory with a vertical action $mv(z)$, a sensing action $sonar(z)$ and two horizontal actions: *towards* moves the robot halfway towards the wall and *away* moves the robot halfway away from the wall. Formally, let $\mathcal{D}_0 \cup \Sigma$ be an action theory where $\mathcal{D}_0$ contains just (6), and Σ includes

- foundational axioms and (P1)-(P3) as usual;
- a l-axiom for $sonar(z)$, namely (13);
- l-axioms for the other actions, which are noise-free, and so these simply equal 1;
- a successor state axiom for v, namely (17);
- the following successor state axiom for h:

$$\begin{aligned} &h(do(a, s)) = \\ &\quad \text{IF } a = away \text{ THEN } 3/2 \cdot h(s) \\ &\quad \text{ELSE IF } a = towards \text{ THEN } 1/2 \cdot h(s) \\ &\quad \text{ELSE } h(s). \end{aligned}$$

We noted that (1) does not satisfy our invertibility property. This variant, however, is invertible. The H-expression for v was derived in Example 15. The H-expression for h is:

$$\begin{aligned} &H_h(a) = \\ &\quad \text{IF } a = away \text{ THEN } 2/3 \cdot h \\ &\quad \text{ELSE IF } a = towards \text{ THEN } 2 \cdot h \\ &\quad \text{ELSE } h. \end{aligned}$$

We now consider the progression of $\mathcal{D}_0$ wrt the action *away*. First, the instantiated H-expressions would simplify to:

- $H_h(away) = 2/3 \cdot h$;
- $H_v(away) = v$.

Next, since *away* is noise-free, we have $Err_{away} = 1$. Putting this together, we obtain $\mathcal{D}_0' = Pro(\mathcal{D}_0, away)$ as:

$$
\begin{aligned}
p(s, S_0) &= \mathcal{U}(2/3 \cdot h; 2, 12) \times \mathcal{N}(v; 0, 1)\ [s] \\
&= \mathcal{U}(h; 3, 18) \times \mathcal{N}(v; 0, 1)\ [s]
\end{aligned}
$$

That is, the new p is one where h is uniformly distributed on $[3, 18]$ and v is independently drawn from a standard normal distribution (as before). This leads to a *shorter* and *wider* density function, as depicted in Figure 4. Here are three simple properties to contrast the original *vs.* the progressed:

- $\mathcal{D}_0 \cup \Sigma \models Bel(h \geq 9, S_0) = .3$.

  The *Bel* term expands as:

  $$
  \frac{1}{\gamma} \int_x \int_y \begin{array}{l} \text{IF } \exists\iota(h = x \wedge v = y \wedge h \geq 9)[\iota] \\ \qquad \text{THEN } \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; 0, 1)[\iota] \ \text{ ELSE } 0 \end{array}
  $$

  which simplifies to the integration of a density function:

  $$
  \begin{aligned}
  & \frac{1}{\gamma} \int_x \int_y \begin{cases} .1 \times \mathcal{N}(y; 0, 1) & \text{if } x \in [2, 12], x \geq 9 \\ 0 & \text{otherwise} \end{cases} \\
  = & \frac{1}{\gamma} \int_x \int_y \begin{cases} .1 \times \mathcal{N}(y; 0, 1) & \text{if } x \in [9, 12] \\ 0 & \text{otherwise} \end{cases} \\
  = & .3.
  \end{aligned}
  $$

  Only those situations where $h \in [2, 12]$ initially are given non-zero p values and by the formula in the *Bel*-term, only those where $h \geq 9$ are to be considered.

- $\mathcal{D}_0 \cup \Sigma \models Bel(h \geq 9, do(away, S_0)) = .6$.

  For any initial situation ι, $h[do(away, \iota)] \geq 9$ only when $h[\iota] \geq 6$, which is given an initial belief of .6.

- $\mathcal{D}_0' \cup \Sigma \models Bel(h \geq 9, S_0) = .6$.

  Basically, *Bel* simplifies to an expression of the form:

  $$
  \frac{1}{\gamma} \int_x \int_y \begin{cases} 1/15 \cdot \mathcal{N}(y; 0, 1) & \text{if } x \in [3, 18], x \geq 9 \\ 0 & \text{otherwise} \end{cases}
  $$

  giving us .6.

Example 26. Let $\mathcal{D}_0 \cup \Sigma$ be exactly as above, and consider its progression wrt *towards*. It is easy to verify that for instantiated H-expressions we get:

- $H_h(towards) = 2 \cdot h$;
- $H_v(towards) = v$;

Here too, because *towards* is noise-free, $Err_{towards}$ is simply 1, which is to say the $\mathcal{D}_0' = Pro(\mathcal{D}_0, towards)$ is defined as:

$$
\begin{aligned}
p(s, S_0) &= \mathcal{U}(2 \times h; 2, 12) \times \mathcal{N}(v; 0, 1)\ [s] \\
&= \mathcal{U}(h; 1, 6) \times \mathcal{N}(v; 0, 1)\ [s].
\end{aligned}
$$

The new distribution on h is *narrower* and *taller*, as shown in Figure 5. Here we might contrast $\mathcal{D}_0$ and $\mathcal{D}_0'$ as follows:

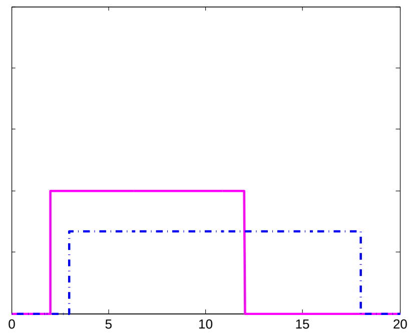

Figure 4: Belief change about h: initially (solid magenta) and after moving away (dotted blue).

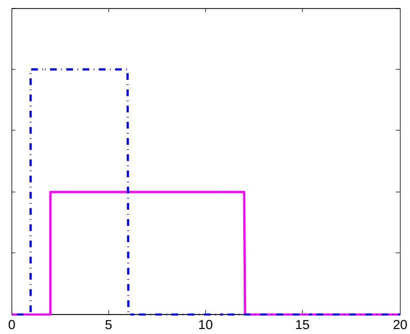

Figure 5: Belief change about h: initially (solid magenta) and moving towards the wall (dotted blue).

- $\mathcal{D}_0 \cup \Sigma \models Bel(h \in [2,3], S_0) = .1$.
- $\mathcal{D}_0' \cup \Sigma \models Bel(h \in [2,3], S_0) = .2$.

Example 27. Let $\mathcal{D}_0 \cup \Sigma$ be as in the previous examples. Consider its progression wrt the action *sonar*(5). Sensing actions do not affect fluents, so for H-expressions we have:

- $H_h(sonar(5)) = h$;
- $H_v(sonar(5)) = v$.

Here $sonar(z)$ is noisy, and we have $Err_{sonar}(5) = \mathcal{N}(5; h, 4)$. This means that the progression $\mathcal{D}_0' = Pro(\mathcal{D}_0, sonar(5))$ is

$$\frac{p(\iota, S_0)}{\mathcal{N}(5; h, 4)[\iota]} = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; 0, 1)[\iota],$$

which simplifies to the following:

$$p(\iota, S_0) = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; 0, 1) \times \mathcal{N}(5; h, 4)\ [\iota].$$

As can be noted in Figure 6, the robot's belief about h's true value around 5 has sharpened. Consider, for example, that:

- $\mathcal{D}_0 \cup \Sigma \models Bel(h \leq 9, S_0) = .7.$
- $\mathcal{D}'_0 \cup \Sigma \models Bel(h \leq 9, S_0) \approx .97.$

If we were to progress $\mathcal{D}'_0$ further wrt a second sensing action, say $sonar(5.9)$, we would obtain the following:

$$
\begin{aligned}
&p(\iota, S_0) = \\
&\quad \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; 0, 1) \times \mathcal{N}(5; h, 4) \times \mathcal{N}(5.9; h, 4)\ [\iota].
\end{aligned}
$$

As can be seen in Figure 6, the robot's belief about h would sharpen significantly after this second sensing action. If we let $\mathcal{D}''_0 = Pro(\mathcal{D}'_0, sonar(5.9))$ then:

- $\mathcal{D}''_0 \cup \Sigma \models Bel(h \leq 9, S_0) \approx .99.$

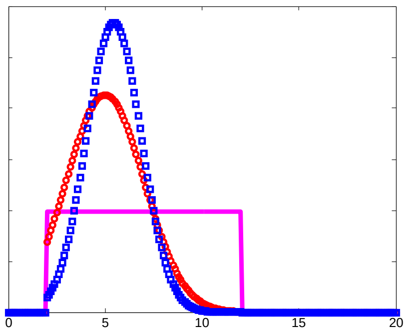

Figure 6: Belief change about h: initially (solid magenta), after sensing 5 (red circles), and after sensing twice (blue squares).

Example 28. Let $\mathcal{D}_0$ be (6). Let Σ be the union of:

- (P1)-(P3) and domain-independent foundational axioms;
- a successor state axiom for h as above;
- a noisy move action mv with the following l-axiom:

$$l(mv(x, y), s) = \mathcal{N}(y; x, 2)$$

- a successor state axiom for v using this noisy move:

$$
\begin{aligned}
v(do(a, s)) = \ &\text{IF } \exists x, y(a = mv(x, y)) \\
&\text{THEN } v(s) + y \text{ ELSE } v(s).
\end{aligned}
$$

  (Recall that for a noisy move $mv(x, y)$, x is the intended motion and y is the actual motion.) This is inverted using the same idea as in Example 15.

Consider the progression of $\mathcal{D}_0 \cup \Sigma$ wrt $mv(2, 3)$, where, of course, the agent does not get to observe the latter argument, and so corresponds to the action $mv(2, u)$. The simplified H-expressions are as follows:

- $H_h(mv(2, u)) = h;$
- $H_v(mv(2, u)) = v - u.$

By definition, occurrences of v in $\mathcal{D}_0$ are to be replaced by $H_v(mv(2,u))$. Also, $Err_{mv}(2,u) = \mathcal{N}(u; 2, 2)$. Therefore, $\mathcal{D}_0' = Pro(\mathcal{D}_0, mv(2,u))$ is defined to be

$$\left(\frac{p(\iota, S_0)}{\mathcal{N}(u; 2, 2)[\iota]} = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v - u; 0, 1)[\iota]\right)$$

This simplifies to:

$$p(\iota, S_0) = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; u, 1) \times \mathcal{N}(u; 2, 2)\ [\iota].$$

Thus the noisy action has had the effect that the belief about the position has shifted by an amount u drawn from a normal distribution centered around 2. This leads to a *shifted* and *wider* curve seen in Figure 7. As expected, the agent is considerably less confident about its position after a noisy move. Here, for example, are the degrees of belief about being located within 1 unit of the best estimate (that is, the mean):

- $\mathcal{D}_0 \cup \Sigma \models Bel(v \in [-1, 1], S_0) \approx .68$.
- $\mathcal{D}_0' \cup \Sigma \models Bel(v \in [1, 3], S_0) \approx .34$.

  Basically, *Bel* expands to an expression of the form

  $$\frac{1}{\gamma} \int_{x,y,z} \begin{cases} .1\mathcal{N}(y; z, 1) \cdot \mathcal{N}(z; 2, 2) & \text{if } x \in [2, 12], y \in [1, 3] \\ 0 & \text{otherwise} \end{cases}$$

  where γ is

  $$\int_{x,y,z} \begin{cases} .1\mathcal{N}(y; z, 1) \cdot \mathcal{N}(z; 2, 2) & \text{if } x \in [2, 12] \\ 0 & \text{otherwise} \end{cases}$$

  leading to .34.

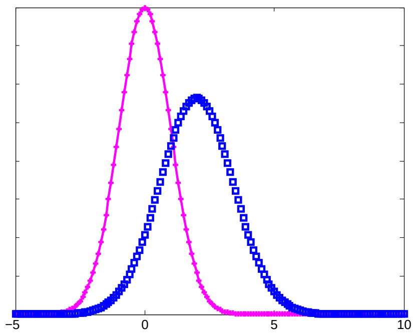

Figure 7: Belief change about v: initially (solid magenta) and after a noisy move of 2 units (blue squares).

8. Computability of Progression

In the general case [47], the computability of progression is a major concern, as it requires second-order logic. We are treating a special case here, and because it is defined over simple syntactic transformations, we have the following result immediately:

Theorem 29. *Suppose $\mathcal{D} = \mathcal{D}_0 \cup \Sigma$ is any invertible basic action theory. After the iterative progression of $\mathcal{D}_0 \cup \Sigma$ wrt a sequence δ, the size of the new initial theory is $O(|\mathcal{D}| \times |\delta|)$.*

Proof: The result of progression is a theory $\mathcal{D}_0'$ which is essentially obtained by means of H-expressions applied to $\mathcal{D}_0$: that is, each fluent appearing in $\mathcal{D}_0$ is replaced by its corresponding H-expression. Note that in the worst-case, H-expressions are of the same size as the successor state axioms, that is, the size is $|\mathcal{D} - \mathcal{D}_0|$. (For example, imagine a basic action theory with a single fluent h, other than p, and a trivial precondition axiom; then the size of $\mathcal{D} - \mathcal{D}_0$ is determined by the successor state axiom for h. Since the H-expression for h inverts the successor state axiom, its size is also $|\mathcal{D} - \mathcal{D}_0|$.) The progressed initial theory is defined as $\mathcal{D}_0$ but with syntactic substitutions for the fluents based on the H-expressions and the likelihood axioms. Thus, after one action, in the worst case, the size of the initial theory is the size of $\mathcal{D}_0$, plus the size of the substitutions ($|\mathcal{D} - \mathcal{D}_0|$); in other words, the size is $|\mathcal{D}|$. Consequently, after $|\delta|$ actions, the size is $|\mathcal{D}| \times |\delta|$. ∎

Therefore, progression is computable. But for realistic robotic applications, even this may not be enough, especially over millions of actions. Consider, for example, that to calculate a degree of belief it will be necessary to integrate the numerical expression for p. What we turn to in this section is a special case that would guarantee that over any length of action sequences, the size of the progressed theory does not change beyond a constant factor. It will use the notion of *context-completeness* [49] and a few simplification rules.

Definition 30. Suppose $F \subseteq \{f_1, \ldots, f_k\}$ is any set of fluents, and $\mathcal{D}_0 \cup \Sigma$ is any invertible basic action theory. We say that $\mathcal{D}_0$ is *complete* wrt F if for any $\phi \in \mathcal{L}_F$, either $\mathcal{D}_0 \models \phi$ or $\mathcal{D}_0 \models \neg\phi$, where $\mathcal{L}_F$ is the sublanguage of $\mathcal{L}$ restricted to the fluents in F.

Definition 31. An invertible basic action theory $\mathcal{D}_0 \cup \Sigma$ is said to be *context-complete* iff

- for every fluent f, $\mathcal{D}_0$ is complete wrt every fluent other than f appearing in the successor state axiom of f;
- $\mathcal{D}_0$ is complete wrt every fluent appearing in a conditional expression in the likelihood axioms.

That is, there is sufficient information in $\mathcal{D}_0$ to simplify all the conditionals appearing in the context formulas of the successor state axioms and the likelihood axioms.

STRIPS actions are trivially context-complete, and so are Reiter's *context-free* successor state axioms where only rigid symbols appear in the RHS[58]. In Example 18, if $\mathcal{D}_0$ is complete wrt the fluents *rate* and *relief*, then the theory would be context-complete. Note that $\mathcal{D}_0$ does not need to be complete wrt the fluent v in that example, and this is precisely why they are interesting. Indeed, both (1) and (17) are also context-complete because, by definition, E_f may mention f, and (say) use its previous value.[16] The reader may further verify that all the density change examples developed in the paper are context-complete. Since we are interested in iterated progression, we say that the progression of $\mathcal{D}$ over an action sequence $\delta = [a_1, \ldots, a_k]$ is context-complete iff *the iterated progression is context-complete*: that is, $\mathcal{D}_0 \cup \Sigma$ is context-complete, $Pro(\mathcal{D}_0, a_1) \cup \Sigma$ is context-complete, $Pro(Pro(\mathcal{D}_0, a_1), a_2) \cup \Sigma$, and so on.

Putting it all together, in contrast to our previous theorem, we would have a progressed theory that in linear in the size of the initial theory $\mathcal{D}_0$:[17]

Theorem 32. *Suppose $\mathcal{D}_0 \cup \Sigma$ is any invertible basic action theory that is also context-complete. After the iterative progression of $\mathcal{D}_0 \cup \Sigma$ wrt a sequence δ, the size of the new initial theory is $O(|\mathcal{D}_0| + |\delta|)$.*

Proof: As a result of context-completeness, the body of the successor state axiom for a fluent f would simplify to an expression involving only f (and one additional free variable resulting from noisy actions). As a consequence, H-expressions simplify to expressions of the form:

$$f(s) = \theta(f(do(a, s)))$$

where $\theta(x)$ is an $\mathcal{L}$-term, possibly an arithmetic expression, with a single free variable x in the case of a noisy action. Thus, for all fluents f in $\mathcal{D}_0$ other than p, the expressions are of the same length, with the addition of some numeric terms and some free variables yielding the form $f \circ_1 n \circ_2 y$, where $\circ_i \in \{+, -, \times, /\}$ and n is a number.

[16] Strictly speaking, our notion of context-completeness is inspired by, but not the same as the one in [49]. This pertains to the clarification just made: we allow successor state axioms to use the fluent's previous value.

[17] A previous version of this theorem [8] bounded the worst case by $|\mathcal{D}_0| \times |\delta|$; the below result shows that it is easily tightened.

The p fluent is affected by the likelihood functions. But analogously, if $A(\vec{t})$ is any ground action, then,

$$Err_A(\vec{t}) = \omega(\vec{t}, \vec{f}).$$

where $\omega(\vec{x}, \vec{y})$ is an $\mathcal{L}$-term, possibly an arithmetic expression, with free variables $\vec{x} \cup \vec{y}$. Such a term is multiplied with p. Thus, the growth in the size of the theory after iterated progression is $|\delta|$. ∎

This can make a substantial difference in the size of the expression for p. A special case of this theorem is immediately applicable to *conjugate distributions* [16, 12], previously considered in Section 5. Indeed, such distributions admit an effective propagation model, as seen in Kalman filtering. We show a simple example where analogous expressions are obtained by our definition of progression:

Example 33. Let $\mathcal{D}_0 \cup \Sigma$ be as in Example 28. We noted its progression wrt $mv(2, 3)$ includes:

$$p(\iota, S_0) = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; z, 1) \times \mathcal{N}(z; 2, 2)\ [\iota].$$

If we progress this sentence further wrt a second noisy action $mv(3, 4)$, we would obtain:

$$p(\iota, S_0) = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v - u; y, 1)\ \times \mathcal{N}(y; 2, 2) \times \mathcal{N}(u; 3, 2)\ [\iota].$$

This then simplifies to:

$$p(\iota, S_0) = \mathcal{U}(h; 2, 12) \times \mathcal{N}(v; y + u, 1)\ \times \mathcal{N}(y; 2, 2) \times \mathcal{N}(u; 3, 2)\ [\iota].$$

9. Related Work

To the best our knowledge, this work is the first to fully generalize classical first-order regression and progression for degrees of belief, noisy acting and sensing. Below, we discuss related efforts in terms of the language, and the individual techniques. We note that although we restricted $\mathcal{L}$ to nullary real-valued fluents, we suspect that both regression and invertibility and its connection to progression may apply more generally. This is left for future investigations.

At the outset, our results were based on a logical language for reasoning about action [58], and its extension to probabilistic beliefs and noise by Bacchus, Halpern and Levesque [2]. Our own prior work further extended that account to continuous distributions [6]. We refer interested readers to [10] for a comprehensive discussion on related efforts on such languages, but summarize the main points below.

The unification of logic and probability has a long history in AI, going back to efforts such as [53]. The works of Bacchus and Halpern [26, 1], in particular, provide the means to specify properties about the domain together with probabilities about propositions; see [54] for a recent list on first-order accounts of probability. The interaction between probability and knowledge was first discussed in [26]. The Bacchus, Halpern and Levesque scheme is closely related to these efforts; see [2] for discussions.

From the perspective of dynamical systems, closest in spirit to our work here are knowledge representation languages for reasoning about action and knowledge, which we refer to as action logics. The situation calculus [51, 58], which has been the sole focus of this paper, is one such language. There are others, of course, such as the event calculus [39], dynamic logic [73, 74], the action language [30], or the fluent calculus [68]. Probabilistic planning languages planning languages [77, 61, 40] are also related on this front. In [38], dynamic logic is extended for knowledge, subjective probabilities and stochastic outcomes. Extending that framework, the notion of updating epistemic states in the presence of announcements and nested beliefs has been studied more comprehensively in [71, 22, 23]. Treating continuous probabilities is also considered for that family of logics in [60].

Limited versions of probabilistic logics are very popular in mainstream AI and machine learning, in the form of relational graphical models or similar [37, 31, 56, 24, 57, 66, 64, 52]. These give up the expressivity of the more powerful languages above for the sake of decidability, or to explore issues such as tractable inference and learnability. Naturally, the generality of our theorems also means we can adapt our results to less expressive initial knowledge bases: in [9], for example, we implemented the regression operator as applicable to a unique joint distribution, such as a Bayesian network.

9.1. *On Regression*

In essence, regression in stochastic domains is addressing the problem of assimilating sensor and effector noise. In standard probabilistic frameworks, perhaps the most popular model to treat sensor fusion is Kalman filtering [28, 69], where priors and likelihoods are assumed to be Gaussian. We already pointed out some instances of Kalman filtering in our example. Where we differ is that backward chaining is possible even when: (a) no assumptions about the nature of distributions, nor about how distributions and dependencies change need to be made, (b) the framework is embedded in a rich theory of actions, and (c) arbitrary forms of incomplete knowledge are allowed, including strict uncertainty. Domain-specific dependencies, then, may be exploited as appropriate.

There is one other thread of related work, that of *symbolic dynamic programming* [15, 14]. (In that vein, algorithms for partially observable Markov decision processes [35] are similar in spirit, their computation is based on expected discount reward measures, and a factorized belief state that is approximated.) While regression is used in this literature as well, the concerns are very different: they focus on policy generation, while ours is strictly about *belief change*. Consequently, the regression in that literature is adapted from the regression for the non-epistemic situation calculus [58]. Ours, on the other hand, continues in the tradition of the epistemic situation calculus [62] by extending those intuitions to probabilistic belief and noisy sensing. In this regard, our account allows the modeler to explicitly reason about beliefs in the language, which would prove useful in formalizing the achievability of plans [44], among other issues.

Of course, as discussed earlier, the idea of regression is not new and lies at the heart of many planning systems [29]. For STRIPS actions, regression has at most linear complexity in the length of the action sequence [58]. For other studies, see [72, 59]. Independently of our efforts, [36] also explore a regression operator for stochastic belief states. They do not, however, consider any correctness formulation, and do not consider first-order logics of probability.

9.2. *On Progression*

Our work on progression builds on Lin and Reiter's [47] account. Other advances on progression have been made since then [48, 75], mainly by appealing to the notion of *forgetting* [46]. We were motivated by concerns in stochastic domains, and this led to the notion of invertible theories. These theories allowed us to perform first-order progression by inverting successor state axioms in a way that, as far as we know, has not been investigated before, even in non-stochastic settings.

The progression of categorical knowledge against noise-free effectors and sensors is considered in [50, 41]. The progression of discrete degrees of belief wrt context-completeness is considered in [4]. In the fluent calculus [67], a dual form of successor state axioms is used, leading naturally to a form of progression. However, continuity is not considered in any of these.

The form of progression considered here follows Lin and Reiter and differs from weaker forms including the one proposed by Liu and Levesque [49], and the notion of *logical filtering* [63, 33], which is a form of (approximate) progression.[18] Interestingly, logical filtering is inspired by Kalman filters [69], although the precise connection is not considered. In situation calculus terminology, Kalman filters and its variants are derived using *strongly* context-free [58] noisy actions and sensors, with additive Gaussian noise, over normally distributed fluents. So they are a special case of the more general account we developed.

10. Conclusions

Planning and robotic applications have to deal with numerous sources of complexity regarding action and change. Along with efforts in related knowledge representation formalisms such as dynamic logic [73], the action language [30] and the fluent calculus [68], Reiter's [58] reconsideration of the situation calculus has proven enormously useful for the design of logical agents, essentially paving the way for *cognitive robotics* [42].

In this work, we obtained new results on how to handle projection in the presence of probabilistic information, both at the level of the knowledge base and at the level of actions. In particular, we generalized both regression and progression.

[18] In [63, 33], the notion of a *permuting action* is introduced for computing their form of progression, which bears some similarity to invertible fluents. However, as mentioned above, our work on continuity led itself to invertibility. Neither continuous uncertainty nor continuous noise is considered in [63, 33].

Our regression results are interesting because irrespective of the decompositions and factorizations that are justifiable initially, belief state evolution is known to invalidate these factorizations even over simple temporal phenomena. We demonstrated regression in settings where actions affect priors in nonstandard ways, such as transforming a continuous distribution to a mixed one. In general, regression does not insist on (but allows) restrictions to actions, that is, no assumptions need to be made about how actions affect variables and their dependencies over time. Moreover, at the specification level, we do not assume (but allow) structurally constrained initial states.

Given the generality of our results, and the promising advances made in the area of relational probabilistic inference [18], we believe regression suggests natural ways to apply those developments with actions. This line of research would allow us to address effective belief propagation for numerous planning problems that require both logical and probabilistic representations. On another front, note that after applying the reductions, one may also use approximate inference methods. Perhaps then, regression can serve as a computational framework to study approximate belief propagation, on the one hand, and using approximate inference at the initial state after goal regression, on the other. As discussed, [9] already implements the regression operator over any single probability distribution, including factorized representations such as Bayesian networks, which is evaluated by sampling.

With regards to progression, Lin and Reiter developed their notion with long-lived agents in mind. However, their account did not deal with probabilistic uncertainty nor with noise, as seen in real-world robotic applications. In the work here, we consider semantically correct progression in the presence of continuity. By first identifying what we called invertible basic action theories, we obtained a new way of computing progression. Under the additional restriction of context-completeness, progression is very efficient. Most significantly, by working within a richer language, we have obtained progression machinery that, to the best of our knowledge, has not been discussed elsewhere, and goes beyond existing techniques. The unrestricted nature of the specification of the p fluent, for example, which we inherit from [2], allows for agents whose beliefs are not determined by a unique distribution. There are two immediate directions for future work on progression. First, just like regression was implemented in [9], it would be worthwhile to investigate an implementation for progression. Second, the invertibility property was mainly sought to handle continuity, including the case where a continuous distribution transforms to a discrete one. If we restrict our attention to discrete distributions, the natural question is whether one can obtain an account of progression in stochastic domains that does not syntactically restrict the basic action theory.

Finally, building on both our results, developing an epistemic planner that leverages the ideas behind regression and progression, as one would in classical planning, would make for very interesting future work. Extending such a framework to multiple agents would also be an exciting future direction.

[1] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press, 1990.

[2] 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.

[3] C. Baral, T. Bolander, H. van Ditmarsch, and S. McIlrath. Epistemic Planning (Dagstuhl Seminar 17231). *Dagstuhl Reports*, 7(6):1–47, 2017.

[4] V. Belle and G. Lakemeyer. A semantical account of progression in the presence of uncertainty. In *Proc. AAAI*, pages 165–170, 2011.

[5] V. Belle and G. Lakemeyer. Reasoning about probabilities in unbounded first-order dynamical domains. In *IJCAI*, 2017.

[6] V. Belle and H. J. Levesque. Reasoning about continuous uncertainty in the situation calculus. In *Proc. IJCAI*, 2013.

[7] V. Belle and H. J. Levesque. Reasoning about probabilities in dynamic systems using goal regression. In *Proc. UAI*, 2013.

[8] V. Belle and H. J. Levesque. How to progress beliefs in continuous domains. In *Proc. KR*, 2014.

[9] V. Belle and H. J. Levesque. PREGO: An Action Language for Belief-Based Cognitive Robotics in Continuous Domains. In *Proc. AAAI*, 2014.

[10] V. Belle and H. J. Levesque. Reasoning about discrete and continuous noisy sensors and effectors in dynamical systems. *Artificial Intelligence*, 262:189–221, 2018.

[11] V. Belle, A. Passerini, and G. Van den Broeck. Probabilistic inference in hybrid domains by weighted model integration. In *IJCAI*, 2015.

[12] D. Bertsekas and J. Tsitsiklis. *Introduction to probability*. Athena Scientific, 2nd edition edition, 2008.

[13] K. Binmore and J. Davies. *Calculus: concepts and methods*. Cambridge University Press, 2002.

[14] C. Boutilier, R. Reiter, and B. Price. Symbolic dynamic programming for first-order MDPs. In *Proc. IJCAI*, pages 690–697, 2001.

[15] C. Boutilier, R. Reiter, M. Soutchanski, and S. Thrun. Decision-theoretic, high-level agent programming in the situation calculus. In *Proc. AAAI*, pages 355–362, 2000.

[16] G. E. P. Box and G. C. Tiao. *Bayesian inference in statistical analysis*. Addison-Wesley, 1973.

[17] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proc. UAI*, pages 33–42, 1998.

[18] L. De Raedt and K. Kersting. Statistical relational learning. In *Encyclopedia of Machine Learning*, pages 916–924. Springer, 2011.

[19] T. Dean and K. Kanazawa. Probabilistic temporal reasoning. In *Proc. AAAI*, pages 524–529, 1988.

[20] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2):142–150, 1989.

[21] T. Dean and M. Wellman. *Planning and control*. Morgan Kaufmann Publishers Inc., 1991.

[22] L. Demey and B. Kooi. Logic and probabilistic update. In *Johan van Benthem on Logic and Information Dynamics*, pages 381–404. Springer, 2014.

[23] L. Demey and J. Sack. Epistemic probabilistic logic. *Handbook of Epistemic Logic*, 2015.

[24] P. Domingos and W. A. Webb. A tractable first-order probabilistic logic. In *AAAI*, 2012.

[25] H. Enderton. *A mathematical introduction to logic*. Academic press New York, 1972.

[26] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994.

[27] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.

[28] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *Pervasive Computing, IEEE*, 2(3):24–33, 2003.

[29] C. Fritz and S. A. McIlraith. Monitoring plan optimality during execution. In *Proc. ICAPS*, pages 144–151, 2007.

[30] M. Gelfond and V. Lifschitz. Special issue: Non-monotonic reasoning and logic programming representing action and change by logic programs. *The Journal of Logic Programming*, 17(2):301 – 321, 1993.

[31] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *ICML*, pages 170–177, 2001.

[32] V. Gogate and P. Domingos. Formula-based probabilistic inference. In *Proc. UAI*, pages 210–219, 2010.

[33] H. Hajishirzi and E. Amir. Reasoning about deterministic actions with probabilistic prior and application to stochastic filtering. In *Proc. KR*, 2010.

[34] J. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311–350, 1990.

[35] 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.

[36] 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.

[37] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *AAAI/IAAI*, pages 580–587, 1998.

[38] B. Kooi. Probabilistic dynamic epistemic logic. *Journal of Logic, Language and Information*, 12(4):381–408, 2003.

[39] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[40] N. Kushmerick, S. Hanks, and D. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76(1):239–286, 1995.

[41] G. Lakemeyer and H. Levesque. A semantical account of progression in the presence of defaults. In *Conceptual Modeling: Foundations and Applications*, pages 82–98. Springer, 2009.

[42] G. Lakemeyer and H. J. Levesque. Cognitive robotics. In *Handbook of Knowledge Representation*, pages 869–886. Elsevier, 2007.

[43] H. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. Scherl. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming*, 31:59–84, 1997.

[44] H. J. Levesque. What is planning in the presence of sensing? In *Proc. AAAI / IAAI*, pages 1139–1146, 1996.

[45] H. J. Levesque, F. Pirri, and R. Reiter. Foundations for the situation calculus. *Electron. Trans. Artif. Intell.*, 2:159–178, 1998.

[46] F. Lin and R. Reiter. Forget it. In *Working Notes of AAAI Fall Symposium on Relevance*, pages 154–159, 1994.

[47] F. Lin and R. Reiter. How to progress a database. *Artificial Intelligence*, 92(1-2):131–167, 1997.

[48] Y. Liu and G. Lakemeyer. On first-order definability and computability of progression for local-effect actions and beyond. In *Proc. IJCAI*, pages 860–866, 2009.

[49] Y. Liu and H. Levesque. Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions. In *Proc. IJCAI*, pages 522–527, 2005.

[50] Y. Liu and X. Wen. On the progression of knowledge in the situation calculus. In *IJCAI*, 2011.

[51] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence*, pages 463–502, 1969.

[52] B. Milch, B. Marthi, S. J. Russell, D. Sontag, D. L. Ong, and A. Kolobov. BLOG: Probabilistic models with unknown objects. In *Proc. IJCAI*, pages 1352–1359, 2005.

[53] N. J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[54] Z. Ognjanovic and M. Raškovic. Some first-order probability logics. *Theoretical Computer Science*, 247(1–2):191 – 212, 2000.

[55] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[56] D. Poole. First-order probabilistic inference. In *Proc. IJCAI*, pages 985–991, 2003.

[57] L. D. Raedt, A. Kimmig, and H. Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proc. IJCAI*, pages 2462–2467, 2007.

[58] R. Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.

[59] J. Rintanen. Regression for classical and nondeterministic planning. In *Proc. ECAI*, pages 568–572, 2008.

[60] J. Sack. Extending probabilistic dynamic epistemic logic. *Synthese*, 169(2):241–257, 2009.

[61] S. Sanner. Relational dynamic influence diagram language (rddl): Language description. Technical report, Australian National University, 2011.

[62] R. B. Scherl and H. J. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1-2):1–39, 2003.

[63] A. Shirazi and E. Amir. First-order logical filtering. In *Proc. IJCAI*, pages 589–595, 2005.

[64] P. Singla and P. M. Domingos. Markov logic in infinite domains. In *UAI*, pages 368–375, 2007.

[65] R. Smullyan. *First-order logic*. Dover Publications, 1995.

[66] D. Suciu, D. Olteanu, C. Ré, and C. Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.

[67] M. Thielscher. Planning with noisy actions (preliminary report). In *Proc. Australian Joint Conference on Artificial Intelligence*, pages 27–45, 2001.

[68] M. Thielscher. *Reasoning robots: the art and science of programming robotic agents*. Applied logic series. Springer, Dordrecht, 2005.

[69] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[70] W. F. Trench. *Introduction to real analysis*. Prentice Hall, 2003.

[71] J. Van Benthem, J. Gerbrandy, and B. Kooi. Dynamic update with probabilities. *Studia Logica*, 93(1):67–96, 2009.

[72] H. Van Ditmarsch, A. Herzig, and T. De Lima. Optimal regression for reasoning about knowledge and actions. In *Proc. AAAI*, pages

1070–1075, 2007.
[73] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition, 2007.
[74] H. P. van Ditmarsch, A. Herzig, and T. D. Lima. From situation calculus to dynamic epistemic logic. *J. Log. Comput.*, 21(2):179–204, 2011.
[75] S. Vassos. *A Reasoning Module for Long-Lived Cognitive Agents*. PhD thesis, University of Toronto, Toronto, Canada, 2009.
[76] S. Vassos and H. Levesque. On the Progression of Situation Calculus Basic Action Theories: Resolving a 10-year-old Conjecture. In *Proc. AAAI*, pages 1004–1009, 2008.
[77] H. Younes and M. Littman. PPDDL 1. 0: An extension to pddl for expressing planning domains with probabilistic effects. Technical report, Carnegie Mellon University, 2004.