From knowledge-based programs to graded belief-based programs, part II: off-line reasoning

Noël Laverny
IRIT
Université Paul Sabatier
31062 Toulouse Cedex, France
laverny@irit.fr

Jérôme Lang
IRIT
Université Paul Sabatier
31062 Toulouse Cedex, France
lang@irit.fr

Abstract

Belief-based programs generalize knowledge-based programs [Fagin *et al.*, 1995] by allowing for incorrect beliefs, unreliable observations, and branching conditions that refer to implicit graded beliefs, such as in "`while` my belief about the direction to the railway station is not strong enough `do ask` someone". We show how to reason off-line about the possible executions of a belief-based program, which calls for introducing second-order uncertainty in the model.

1 Introduction

Knowledge-based programs, or KBPs [Fagin *et al.*, 1995] are a powerful notion for expressing action policies in which branching conditions refer to knowledge (an agent acts according to what he knows), such as, typically,

$$\texttt{if } \mathbf{K}\varphi \texttt{ then } \pi \texttt{ else } \pi'$$

where $\mathbf{K}$ is an epistemic (usually S5) modality, and π, π' are subprograms. However, branching conditions in KBPs cannot refer to possibly erroneous beliefs or to graded belief, such as in "`repeat ask` to someone about the way to the railway station `until` my belief about the direction to take is strong enough". Recently, [Laverny and Lang, 2004] made a first step towards reasoning with *belief-based programs* (BBPs), where knowledge modalities of KBPs are replaced by graded belief modalities, whose semantics relies on belief states defined as ranking functions, which can be revised by observations and progressed by physical actions, which enables the agent to maintain his current belief state about the world *while executing* a BBP. Note that BBPs extend a restricted class of KBPs, in which (a) there is a single agent, (b) the agent has perfect recall and (c) branching conditions concern the agent's knowledge about the present state of the world.

However, [Laverny and Lang, 2004] cannot deal with *off-line reasoning about the effects of a belief-based program*. Assume for instance that agent A is looking for the way to the railway station in a foreign town; her initial belief state is void, and she follows a plan consisting in sequentially asking to several pedestrians about the direction to follow, until she has acquired a sufficient level of certainty. Assume moreover that answers, although not fully reliable, are normally correct. Each time A gets a direction confirming (resp. contradicting) her current belief, this belief becomes stronger (resp. weaker). Now, the assumption that answers are normally correct implies, for instance, that if A has already got 3 "right" and no "left" then the next answer is more likely to be "right" again than "left": the plausibility of getting an observation depends on the current state.

Coping with this issue results in a natural way in *second-order uncertainty* when projecting a belief state by a BBP: in our example, the agent is able to predict beforehand something like "after asking 5 pedestrians, normally I'll have a very firm belief about the direction to the station, although I cannot totally exclude the possibility that I'll have only a weak, or even a totally void belief". Such a complex belief state is a belief state about (future) belief states, that is, a second-order belief state. Thus, the main concern of this paper is how to do *a priori* reasoning about the possible states of belief produced by executing the program: uncertainty about which of these states of belief will result is itself represented as a ranking over states of belief.

After recalling some background in Section 2, belief-based programs are introduced in Section 3. Section 4 deals with *complex belief states* and their progression by actions and programs. In Section 5 we show how to compute progression syntactically. Related work is discussed in Section 6.

2 Background

In this Section we briefly recall some notions from [Laverny and Lang, 2004]. Let PS be a *finite* set of propositional symbols. L_{PS} is the (non-modal) propositional language generated from PS, the usual connectives and the Boolean constants $\top$ and $\bot$. $S = 2^{PS}$ is the set of *states* associated with PS. Formulas of L_{PS} are said *objective*. If $\varphi \in L_{PS}$ then $Mod(\varphi) = \{s \in S | s \models \varphi\}$. For $A \subseteq S$, $Form(A)$ is the objective formula (unique up to logical equivalence) such that $Mod(Form(A)) = A$.

Belief states are identified with *ordinal conditional functions* (OCF) [Spohn, 1988]: a belief state is a function $\kappa : S \longmapsto \overline{\mathbb{N}}$, where $\overline{\mathbb{N}} = \mathbb{N} \cup \{+\infty\}$, such that $\min_{s\in S} \kappa(s) = 0$. κ is lifted from S to L_{PS} by $\kappa(\varphi) = \min\{\kappa(s) \mid s \models \varphi\}$, with the convention $\min(\emptyset) = +\infty$ (we will use this convention throughout the paper without recalling it). $\kappa(s)$ can be seen as the *exceptionality degree* of s. In particular, $\kappa(s) = 0$

means that s is a normal state and $\kappa(s) = +\infty$ that s is totally impossible. For any $\varphi \in L_{PS}$, the belief state κ_φ is defined by $\kappa_\varphi(s) = \begin{cases} 0 & \text{if } s \models \varphi \\ +\infty & \text{if } s \models \neg\varphi \end{cases}$. In particular, $\kappa_\top$ is the *void belief state*: $\forall s$, $\kappa_\top(s) = 0$.

Beliefs are expressed syntactically in a graded extension $\mathsf{KD45}_G$ of KD45, whose language $\mathcal{L}_{PS}$ is defined as follows:

(a) if $\varphi \in L_{PS}$ then $\mathbf{B_1}\varphi, \mathbf{B_2}\varphi, \ldots, \mathbf{B}_\infty\varphi$ are in $\mathcal{L}_{PS}$;

(b) if Φ and Ψ in $\mathcal{L}_{PS}$ then $\neg\Phi$, $\Phi \vee \Psi$, $\Phi \wedge \Psi$ in $\mathcal{L}_{PS}$.

Note that $\mathcal{L}_{PS}$ considers only *subjective* and *flat* formulas[1]. Formulas of $\mathcal{L}_{\mathcal{PS}}$ are denoted by capital Greek letters Φ, Ψ etc. while objective formulas are denoted by small Greek letters φ, ψ etc. $\mathbf{B_i}\varphi$ intuitively means that the agent believes φ with strength i. The larger i, the stronger the belief expressed by $\mathbf{B_i}$, and $\mathbf{B}_\infty$ is a *knowledge* (true belief) modality.

The truth of a formula of $\mathcal{L}_{\mathcal{PS}}$ in an belief state κ is defined by:

(a) for φ objective and $i \in \overline{\mathbb{N}}$, $\kappa \models \mathbf{B_i}\varphi$ **iff** $\kappa(\neg\varphi) \geqslant i$;

(b) $\kappa \models \Phi \vee \Psi$ **iff** $\kappa \models \Phi$ or $\kappa \models \Psi$;

(b) $\kappa \models \Phi \wedge \Psi$ **iff** $\kappa \models \Phi$ and $\kappa \models \Psi$;

(c) $\kappa \models \neg\Phi$ **iff** $\kappa \not\models \Phi$.

Thus, $\kappa \models \mathbf{B_i}\varphi$ holds as soon as any countermodel of φ is exceptional at least to the degree i, or, equivalently, that all states such that $\kappa(s) < i$ are models of φ. In particular, $\mathbf{B_1}\varphi$ is satisfied when all normal states satisfy φ, and $\mathbf{B}_\infty\varphi$ is satisfied when all possible states (to any degree) are models of φ.

An *observation* is a belief state κ_{obs}, representing *all we observe* when getting the observation. Observations can be incomplete and partially unreliable (see [Laverny and Lang, 2004] for examples). The agent revises her current belief state by an observation by combining both: the revision of κ by κ_{obs} is undefined when $\min_S(\kappa + \kappa_{obs}) = \infty$, and otherwise is the belief state defined by

$$\forall s \in S, (\kappa \oplus \kappa_{obs})(s) = \kappa(s) + \kappa_{obs}(s) - \min_S(\kappa + \kappa_{obs})$$

In particular, $\kappa_\top \oplus \kappa_{obs} = \kappa_{obs}$ and $(\kappa \oplus \kappa_\varphi) = \kappa(.|\varphi)$, where $\kappa(.|\varphi)$ is Spohn's conditioning [Spohn, 1988].

A physical action α is a feedback-free action (that is, it possibly changes the state of the world but does not give any feedback), defined by a transition model consisting of a collection of belief states $\{\kappa_\alpha(.|s), s \in S\}$. $\kappa_\alpha(s'|s)$ is the exceptionality degree of the outcome s' when performing α in state s. The *progression* of a belief state κ_0 by α is the belief state $\kappa \diamond \alpha = \kappa(.|\kappa_0, \alpha)$ defined (cf. [Boutilier, 1998]) by

$$\forall s \in S, (\kappa \diamond \alpha)(s) = \kappa(s|\kappa_0, \alpha) = \min_{s' \in S}\{\kappa(s') + \kappa_\alpha(s|s')\}$$

A *positive formula* of $\mathcal{L}_{\mathcal{PS}}$ is a formula where no $\mathbf{B_i}$ appears in the scope of negation. A *positive conjunctive* (PC) formula of $\mathcal{L}_{\mathcal{PS}}$ is a formula of the form $\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B_n}\varphi_n \wedge \ldots \wedge \mathbf{B_1}\varphi_1$; without loss of generality we can assume $\varphi_i \models \varphi_{i+1}$, since $\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B_n}\varphi_n \wedge \ldots \wedge \mathbf{B_1}\varphi_1$ is equivalent to $\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B_n}(\varphi_\infty \wedge \varphi_n) \wedge \ldots \wedge \mathbf{B_1}(\varphi_\infty \wedge \varphi_n \wedge \ldots \wedge \varphi_1)$. There is a one-to-one correspondance between belief states and satisfiable PC formulas (modulo logical equivalence): for each κ, the PC formula $G(\kappa) = \Phi_\kappa$ is defined as $\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B_n}\varphi_n \wedge \ldots \wedge \mathbf{B_1}\varphi_1$, where $n = \max\{k < \infty, \exists s \text{ such that } \kappa(s) = k\}$, and for every $i \in \overline{\mathbb{N}}^*$, $\varphi_i = Form(\{s, \kappa(s) < i\})$. (Note that n is finite, because S is finite and $\min \kappa = 0$.) For instance, let $\kappa([a, b]) = 0$, $\kappa([a, \neg b]) = 1$, $\kappa([\neg a, b]) = 3$ and $\kappa([\neg a, \neg b]) = \infty$, then $\Phi_\kappa = \mathbf{B}_\infty(a \vee b) \wedge \mathbf{B_3}a \wedge \mathbf{B_2}a \wedge \mathbf{B_1}(a \wedge b)$ – which is equivalent to $\mathbf{B}_\infty(a \vee b) \wedge \mathbf{B_3}a \wedge \mathbf{B_1}b$. Conversely, for each satisfiable PC formula Ψ there is a belief state $\kappa_\Psi = H(\Psi)$ such that $G(H(\Psi)) \equiv \Psi$. $G(\kappa)$ represents *all the agent believes* in κ[2]. We will sometimes make the following slight abuse of notation: when a PC formula Ψ is equivalent to a shorter (but not PC) formula Ψ', we write $\kappa_{\Psi'}$ instead of κ_Ψ. For instance, the PC formula $\Psi = \mathbf{B}_\infty\varphi_\infty\top \wedge \mathbf{B_3}\top \wedge \mathbf{B_2}r \wedge \mathbf{B_1}r$ is equivalent to $\mathbf{B_2}r$, therefore we write $\kappa_{\mathbf{B_2}r}$ instead of κ_Ψ.

3 Belief-based programs

Belief-based programs (BBP) are built up from a set of actions ACT and program constructors:

- the empty plan λ is a BBP;
- for any $\alpha \in ACT$, α is a BBP;
- if π and π' are BBPs then $(\pi; \pi')$ is a BBP;
- if π and π' are BBP and $\Phi \in \mathcal{L}_{\mathcal{PS}}$, then $(\texttt{if } \Phi \texttt{ then } \pi \texttt{ else } \pi')$ and $(\texttt{while } \Phi \texttt{ do } \pi)$ are BBPs.

Thus, a BBP is a program *whose branching conditions are doxastically interpretable*: the agent can decide whether she *believes* to a given degree that a formula is true (whereas she is generally unable to decide whether a given objective formula is true in the actual world). For instance, the agent performing the BPP

$$\pi = \begin{array}{l} \texttt{while } \neg(\mathbf{B_2}r \vee \neg\mathbf{B_2}\neg r) \texttt{ do ask;} \\ \texttt{if } \mathbf{B_2}r \texttt{ then } goright \texttt{ else } goleft \end{array}$$

performs the sensing action `ask` until she has a belief firm enough (namely of degree 2) about the way to follow (whether π is guaranteed to stop is a good question!).

Progression and revision in [Laverny and Lang, 2004] are used for maintaining the agent's current belief state *while the program is being executed*. However, predicting the future possible states resulting from the execution of a BBP *before* it has started to be executed (*off-line* evaluation) cannot be done in a satisfactory way. Consider $\pi = \texttt{ask}; \texttt{ask}; \texttt{ask}; \texttt{ask}$, consisting in asking in sequence to 4 pedestrians about the way to the station. Assume that each occurrence of `ask` can send

[1]This restriction is made for the sake of simplicity; it would be possible to consider nested modalities, and then prove, as it is the case in KD45, that each formula is equivalent to a flat formula, but this issue has no relevance to the issues dealt with in this paper. Likewise, combinations of objective and subjective formulas do not play any role either for expressing belief-based programs.

[2]This could be formalized by extending our language with graded doxastic versions $\mathbf{O_1}, \ldots, \mathbf{O}_\infty$ of the *only knowing* modality (e.g. [Levesque and Lakemeyer, 2000]), $\mathbf{O_i}\varphi$ meaning that *all the agent believes to the degree* i is φ. Due to space limitations we must omit the details.

back $obs_1 = \kappa_{\mathbf{B_1}r}$ or $obs_2 = \kappa_{\mathbf{B_1}\neg r}$, corresponding respectively to a pedestrian telling that the station is on the right (resp. left), taken with some limited reliability (for the sake of simplicity we exclude "don't know" answers). Then all we can predict is that after doing π the agent will be in one of the 5 belief states $\kappa_{\mathbf{B_4}r}$, $\kappa_{\mathbf{B_4}\neg r}$, $\kappa_{\mathbf{B_2}r}$, $\kappa_{\mathbf{B_2}\neg r}$, $\kappa_{\top}$[3]. The point now is that obs_1 and obs_2 cannot always be considered as likely as each other: for instance, we may wish to express that accurate observations are more frequent than inaccurate ones. Therefore, observations should be ranked by their plausibility of occurrence given the current state and the sensing action performed. Then, the projection of an initial belief state by a program results in a *second-order* (or *complex*) belief state: in our example, one would expect to obtain that after asking to two persons, then normally the agent is the belief state $\kappa_{\mathbf{B_2}r}$ or in the belief state $\kappa_{\mathbf{B_2}\neg r}$, and exceptionally in the void belief state $\kappa_{\top}$. This issue is developed in next Section.

4 Complex belief states and progression

4.1 Complex belief states

Definition 1 *Let B_S be the set of all belief states on S. A* complex belief state *(CBS) is an ordinal conditional function μ on B_S,* i.e., *a function $\mu : \mathcal{B}_S \to \overline{\mathbb{N}}$ such that* $\min_{\kappa \in \mathcal{B}_S} \mu(\kappa) = 0$

μ is a second-order belief state expressing the beliefs, before executing some given program π, about the (future) possible belief states resulting from its execution.

Example 1 *Let $S = \{r, \neg r\}$.*

$$\mu : \left[\begin{array}{lll} \kappa_0 : \left[\begin{array}{c} r : 0 \\ \neg r : 0 \end{array} \right] & : & \mu(\kappa_0) = 1 \\ \\ \kappa_3 : \left[\begin{array}{c} r : 0 \\ \neg r : 2 \end{array} \right] & : & \mu(\kappa_3) = 0 \\ \\ \kappa_4 : \left[\begin{array}{c} r : 2 \\ \neg r : 0 \end{array} \right] & : & \mu(\kappa_4) = 0 \end{array} \right]$$

is a CBS (by convention, for any belief state κ not mentioned we have $\mu(\kappa) = +\infty$); it intuitively represents a situation where the agent expects the resulting belief state to be either κ_0, κ_3 or κ_4, these last two being normal *results and κ_0 being exceptional. Note that $\kappa_0 = \kappa_{\top}$, $\kappa_3 = \kappa_{\mathbf{B_2}r}$ and $\kappa_4 = \kappa_{\mathbf{B_2}\neg r}$.*

We define μ_κ as the (degenerated) CBS defined by $\mu_\kappa(\kappa) = 0$ and $\mu_\kappa(\kappa') = \infty$ for all $\kappa' \neq \kappa$.

Note that since, unlike S, B_S is not finite, some CBSs cannot be finitely represented. We say that μ has a *finite support* iff only a finite number of belief states have a finite plausibility, *i.e.*, $\{\kappa \in B_S | \mu(\kappa) < \infty\}$ is finite; in this case we define n_μ (or simply n where there is no ambiguity) as $\max\{i < \infty \mid \exists \kappa \text{ such that } \mu(\kappa) = i\}$.

4.2 Progression by sensing actions

Consider a finite set ACT_S of *sensing actions*, that send feedback to the agent, under the form of *observations*. Each sensing action is defined by a state-dependent plausibility assignment on possible observations.

[3]The notation κ_Φ has been introduced at the end of Section 2.

Definition 2 *Let $OBS \subseteq B_S$ be a* finite *set of possible observations (recall that observations are belief states). An* observation model *is a collection of functions*

$$\kappa_{OBS}(.|s, \alpha) : OBS \to \overline{\mathbb{N}}$$

for every $\alpha \in ACT_S$ and every $s \in S$, such that:

1. *for every $\alpha \in ACT_S$ and every $s \in S$,* $\min_{obs \in OBS} \kappa_{OBS}(obs|s, \alpha) = 0$
2. *for every $obs \in OBS$, if $obs(s) = \infty$ then for every $\alpha \in ACT_S$, $\kappa_{OBS}(obs|s, \alpha) = \infty$.*

$\kappa_{OBS}(obs|s, \alpha)$ is the exceptionality degree of getting observation obs as feedback when executing the sensing action α in state s. This definition first appears in [Boutilier *et al.*, 1998], and is similar in spirit to correlation probabilities between states and observations in partially observable Markov decision processes. Condition 1 expresses that there is always at least one normal observation; Condition 2 is a weak consistency condition between states and observations (an observation totally excluding a state cannot occur in that state).

Example 2
Let $S = \{r, \neg r\}$, $obs_1 = \kappa_{\mathbf{B_1}r}$, $obs_2 = \kappa_{\mathbf{B_1}\neg r}$, and let

$$\begin{array}{ll} \kappa_{OBS}(obs_1|r, \mathtt{ask}) = 0 & \kappa_{OBS}(obs_2|r, \mathtt{ask}) = 1 \\ \kappa_{OBS}(obs_2|\neg r, \mathtt{ask}) = 0 & \kappa_{OBS}(obs_1|\neg r, \mathtt{ask}) = 1 \end{array}$$

(all other observations being impossible, i.e., for $obs \neq obs_1, obs_2$, $\kappa(obs|s, \alpha) = \infty$; by convention we omit these when specifying $\kappa(.|s, \alpha)$). This means that accurate observations are the normal ones, whereas incorrect obervations are 1-exceptional.

Definition 3 *Let κ_0 be a belief state and $\alpha \in ACT_S$. Given $obs \in OBS$, the plausibility of obtaining obs after α in belief state κ_0 is defined by*

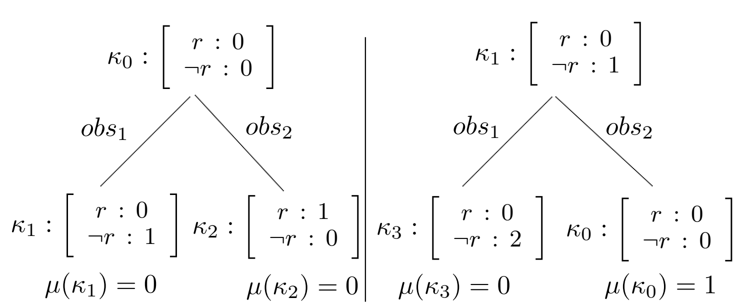
$$\kappa_{OBS}(obs|\kappa_0, \alpha) = \min_{s \in S}[\kappa_0(s) + \kappa_{OBS}(obs|s, \alpha)]$$

The progression *of κ_0 by α is the complex belief state $prog(\kappa_0, \alpha) = \mu(.|\kappa_0, \alpha)$ defined by: for all $\kappa \in \mathcal{B}_S$,*

$$\begin{array}{ll} & \mu(\kappa|\kappa_0, \alpha) \\ = & \min\{\kappa_{OBS}(obs|\kappa_0, \alpha) \mid obs \in OBS \text{ and } \kappa = \kappa_0 \oplus obs\} \end{array}$$

Thus, κ is all the more normal in the projected CBS $\mu(.|\kappa_0, \alpha)$ as there exists a normal state s and a normal observation obs (given s) such that κ is the revision of κ_0 by obs. Condition 2 in Definition 2 guarantees that $\kappa_0 \oplus obs$ is defined whenever $\kappa_{OBS}(obs|\kappa_0, \alpha) < \infty$, which ensures that $\mu(.|\kappa_0, \alpha)$ is a CBS.

Example 3 *The figure on the left (resp. right) shows the progression of κ_0 (resp. κ_1) by* ask.



$$\mu(.|\kappa_0, \texttt{ask}) = \left[\begin{array}{ccc} \kappa_1 & : & 0 \\ \kappa_2 & : & 0 \end{array}\right] \quad \mu(.|\kappa_1, \texttt{ask}) = \left[\begin{array}{ccc} \kappa_3 & : & 0 \\ \kappa_0 & : & 1 \end{array}\right]$$

4.3 Progression by physical actions

In addition to sensing actions we consider a finite set ACT_P of *physical, feedback-free actions*[4]. In Section 2, the progression of a belief state by a physical action was defined *as a belief state*. For the sake of homogeneity, we have now to define it as a CBS: the progression $\mu(.|\kappa_0, \alpha)$ of a belief state κ_0 by $\alpha \in ACT_P$ is defined by $\mu(.|\kappa_0, \alpha) = \mu_{\kappa_0 \diamond \alpha}$, i.e.,

$$\mu(\kappa|\kappa_0, \alpha) = \begin{cases} 0 & \text{if } \kappa = \kappa_0 \diamond \alpha \\ \infty & \text{otherwise} \end{cases}$$

4.4 Progression by belief-based programs

The progression of a belief state κ by a BBP π is the *complex belief state* $\mu(.|\kappa, \pi)$ defined inductively by

- if $\pi = \lambda$ then $\mu(.|\kappa, \pi) = \mu_\kappa$;
- if $\pi = \alpha$ then $\mu(.|\kappa, \pi)$ is defined in Section 4.3 if $\alpha \in ACT_P$ and in Section 4.2 if $\alpha \in ACT_S$;
- if $\pi = (\pi_1; \pi_2)$ then $\mu(\kappa'|\kappa, \pi) = \min_{\kappa'' \in BS}(\mu(\kappa''|\kappa, \pi_1) + \mu(\kappa'|\kappa'', \pi_2))$
- if $\pi = \texttt{if } \Phi \texttt{ then } \pi_1 \texttt{ else } \pi_2$ then $\mu(\kappa'|\kappa, \pi) = \begin{cases} \mu(\kappa'|\kappa, \pi_1) & \text{if } \kappa \models \Phi \\ \mu(\kappa'|\kappa, \pi_2) & \text{otherwise} \end{cases}$
- if $\pi = \texttt{while } \Phi \texttt{ do } \pi'$ then $\mu(\kappa'|\kappa, \pi) = \begin{cases} \mu(\kappa'|\kappa, (\pi'; \pi)) & \text{if } \kappa \models \Phi \\ \mu_\kappa(\kappa') & \text{otherwise} \end{cases}$

If π contains no `while` construct then this definition is well-founded (it can be proved that it results a SBS which moreover does not depend on the order in which the above rules are applied). This is not so with `while` constructs, since the (recursive) definition leads to a fixpoint equation whose solution, when defined, is taken to be its least fixpoint when the latter is a CBS, which is not necessarily the case: consider $\kappa_0 = \kappa_\top$ and $\pi = \texttt{while } \top \texttt{ do ask}$; applying the above definition leads to $\mu(\kappa|\kappa_0, \pi) = \infty$ for all κ, which is not a CBS. Moreover, when $\mu(.|\kappa_0, \pi)$ is defined, it does not necessarily have a finite support.

Example 4

- $\mu(.|\kappa_\top, (\texttt{ask}; \texttt{ask})) = [\kappa_3 : 0\,;\ \kappa_4 : 0\,;\ \kappa_0 : 1]$;
- $\mu(.|\kappa_\top, (\texttt{ask}; \texttt{ask}; \texttt{ask}; \texttt{ask})) = [\kappa_{\mathbf{B_4} r} : 0\,;\ \kappa_{\mathbf{B_4} \neg r} : 0\,;\ \kappa_3 : 1\,;\ \kappa_4 : 1\,;\ \kappa_0 : 2]$;
- $\pi_1 = (\texttt{ask}; \texttt{ask}; \texttt{if } \mathbf{B_2} r \vee \mathbf{B_2} \neg r \texttt{ then } \lambda \texttt{ else ask})$. *Then* $\mu(.|\kappa_\top, \pi_1) = [\kappa_3 : 0\,;\ \kappa_4 : 0;\ \kappa_1 : 1\,;\ \kappa_2 : 1]$;
- $\pi_2 = \texttt{while } \neg(\mathbf{B_2} r \vee \mathbf{B_2} \neg r) \texttt{ do ask}$. *Applying the definition gives a fixpoint equation whose solution is* $\mu(.|\kappa_0, \pi_2) = [\kappa_3 : 0\,;\ \kappa_4 : 0]$.[5]

[4]Such a partition between purely sensing and purely physical actions is usual and does not induce a loss of generality.

[5]The question whether this program can run forever is similar to that of whether it is possible that a fair coin tossed repeatedly always turns up heads, and this is thus related to an OCF version of the law of large numbers.

4.5 Two particular cases

An *unobservable environment* consists of a set of physical actions only ($ACT_S = \emptyset$).

A *fully observable environment* is somewhat harder to define formally because of the separation between physical and sensing actions, which prevents us to say that all actions send a full feedback. To cope with this, we assume that ACT_S contains one action $sense(x)$ for each propositional variable, which returns the truth value of x with full certainty, and we require that in a program, any physical action α should be followed by all actions $sense(x)$ – which means that after α and this sequence of sensing actions, the state is known with full certainty. Any program of this kind is said to be *admissible*. The initial state is also required to be known with certainty.

Proposition 1

1. *in an unobservable environment, for any κ_0 and any program π for which $\mu(.|\kappa_0, \pi)$ is defined, there exists a belief state κ such that $\mu(.|\kappa_0, \pi) = \mu_\kappa$;*
2. *in a fully observable environment, for any belief state κ_0 and any admissible program π such that $\mu(.|\kappa_0, \pi)$ is defined, $\mu(\kappa|\kappa_0, \pi) < \infty$ implies that κ is a precise belief state,* i.e., *$\kappa = \kappa_s$ for some $s \in S$.*

5 Progression: syntactical computation

Applying progression as it is defined above has a prohibitive complexity, since it amounts at iterating over all belief states, states, and observations. In this Section we give a more friendly, syntactical way of computing progression, based on a compact representation of complex belief states.

So as to be able to reason about the resulting belief states, we now introduce a new family of modalities $\mathbf{P_1}$, $\mathbf{P_2}$, …, $\mathbf{P_\infty}$ in addition to $\mathbf{B_1}, \ldots, \mathbf{B_\infty}$. While the $\mathbf{B_i}$ modalities deal with uncertainty (*about the current state*), $\mathbf{P_1}$, $\mathbf{P_2}$, …, $\mathbf{P_\infty}$ deal with second-order uncertainty, *i.e., uncertainty about the projected belief state* (with respect to some program).

$\mathcal{L}^2_{PS}$ is the language defined by:

- if Φ is a formula of $\mathcal{L}_{PS}$ then for all i, $\mathbf{P_i}\Phi$ is a formula of $\mathcal{L}^2_{PS}$;
- if Θ and Θ' are formulas of $\mathcal{L}^2_{PS}$ then $\Theta \wedge \Theta'$, $\Theta \vee \Theta'$ and $\neg\Theta$ are formulas of $\mathcal{L}^2_{PS}$.

Like for $\mathbf{B_i}$ modalities, we need not consider nested $\mathbf{P_i}$ modalities, neither heterogeneous combinations of $\mathbf{P_i}$ and $\mathbf{B_i}$ modalities (resp. objective formulas). Satisfaction of a $\mathcal{L}^2_{PS}$ formula by a CBS is defined by

- $\mu \models \mathbf{P_i}\Phi$ *iff* for all $\kappa \in B_S$, $\kappa \models \neg\Phi$ implies $\mu(\kappa) \geqslant i$;
- $\mu \models \Theta \wedge \Theta'$ *iff* $\mu \models \Theta$ and $\mu \models \Theta'$ (and similarly for the other connectives).

Validity, satisfiability and logical consequence are defined in the usual way. Intuitively, $\mathbf{P_1}\Phi$ means that *all belief states that do not satisfy Φ are exceptional at least to the degree i.*

When reasoning about CBS, we are mainly concerned with inferring *positive formulas* – inferring negative formulas such as $\neg P_i\Phi$ is somewhat derivative. We define a *positive* $\mathcal{L}^2_{PS}$ formula as follows:

- if Φ is a positive $\mathcal{L}_{PS}$ formula and $i \in \overline{\mathbb{N}}$ then $\mathbf{P_i}\Phi$ is a positive $\mathcal{L}^2_{PS}$ formula;
- if Θ_1 and Θ_2 are positive $\mathcal{L}^2_{PS}$ formulas then $\Theta_1 \wedge \Theta_2$ and $\Theta_1 \vee \Theta_2$ is a positive $\mathcal{L}^2_{PS}$ formula.

Moreover, a *canonical* $\mathcal{L}^2_{PS}$ formula is a $\mathcal{L}^2_{PS}$ formula of the form $\Theta = \mathbf{P_1}\Phi_1 \wedge \mathbf{P_2}\Phi_2 \wedge \ldots \wedge \mathbf{P_n}\Phi_n \wedge \mathbf{P}_\infty\Phi_\infty$, where Φ_1, $\ldots$, Φ_n, Φ_∞ are positive $\mathcal{L}_{PS}$ formulas[6].

Given a CBS μ with finite support, the canonical $\mathcal{L}^2_{PS}$ formula $G^+(\mu) = \Theta_\mu$ associated with μ is defined by

$$\Theta_\mu = \bigwedge_{i=1}^{i=n} \mathbf{P_i}\left(\bigvee_{\mu(\kappa)<i} G(\kappa)\right) \wedge \mathbf{P}_\infty\left(\bigvee_{\mu(\kappa)<\infty} G(\kappa)\right)$$

where $n = n_\mu$ and $G(\kappa)$ is the canonical $\mathcal{L}_{PS}$ formula corresponding to κ (cf. Section 2).

Proposition 2 *For any CBS μ with finite support and any positive conjunctive $\mathcal{L}^2_{PS}$ formula Θ, $\mu \models \Theta$ if and only if $\Theta_\mu \models \Theta$, that is, Θ_μ is the strongest positive conjunctive $\mathcal{L}^2_{PS}$ formula satisfied by μ.*

Example 1 (continued)

$$\begin{aligned}\Theta_\mu &= \mathbf{P_1}(\mathbf{B_2}r \vee \mathbf{B_2}\neg r) \wedge \mathbf{P}_\infty(\mathbf{B_2}r \vee \mathbf{B_2}\neg r \vee \mathbf{B}_\infty\top)\\ &\equiv \mathbf{P_1}(\mathbf{B_2}r \vee \mathbf{B_2}\neg r)\end{aligned}$$

We are now in position to give a syntactical characterization of progression.

Definition 4 *Let Φ be a positive conjunctive $\mathcal{L}_{PS}$ formula and α be any action. The progression of Φ by α is the canonical $\mathcal{L}^2_{PS}$ formula $Prog(\Phi, \alpha)$ corresponding to the CBS $\mu(.|\kappa_\Phi, \alpha)$*, i.e.,

$$Prog(\Phi, \alpha) = G^+(\mu(.|\kappa_\Phi, \alpha)) = \Theta_{\mu(.|\kappa_\Phi,\alpha)}$$

We now show how the formula $Prog(\Phi, \alpha)$ can be computed without first generating the corresponding μ.

Proposition 3
Let α be a sensing action and $\Phi = \mathbf{B_1}\varphi_1 \wedge \ldots \wedge \mathbf{B_p}\varphi_p \wedge \mathbf{B}_\infty\varphi$ a positive conjunctive $\mathcal{L}_{PS}$ formula. We define:

- *for any $obs \in OBS$ and $i \in \overline{\mathbb{N}}$, $\psi_{i,obs,\alpha} = Form\{s \in S \mid \mu_{OBS}(obs|s, \alpha) < i\}$;*
- *$X_{i,\Phi,\alpha} = \{obs \in OBS \mid (\varphi_1 \wedge \psi_{i,obs,\alpha}) \vee \ldots \vee (\varphi_i \wedge \psi_{1,obs,\alpha}) \not\equiv \bot\}$;*
- *$X_{\infty,\Phi,\alpha} = \{obs \in OBS \mid \varphi \wedge \psi_{\infty,obs,\alpha} \not\equiv \bot\}$;*
- *n is the largest integer i such that $X_{i,\Phi,\alpha} \subsetneq X_{\infty,\Phi,\alpha}$.*

Then

$$\begin{aligned}Prog(\Phi, \alpha) = \bigwedge_{i=1}^{i=n} \mathbf{P_i}(&\bigvee_{obs \in X_{i,\Phi,\alpha}} \Phi \otimes \Phi_{obs})\\ \wedge \mathbf{P}_\infty(&\bigvee_{obs \in X_{\infty,\Phi,\alpha}} \Phi \otimes \Phi_{obs})\end{aligned}$$

where $\otimes$ is the syntactical revision operator (Proposition 1 of [Laverny and Lang, 2004]), which satisfies $\Phi \otimes \Phi_{obs} = G(H(\Phi) \oplus obs)$.

Example 2 (continued) *Let $\Phi = \mathbf{B_1}r$. We get $X_{1,\Phi,\mathtt{ask}} = \{obs_1\}$, $X_{2,\Phi,\mathtt{ask}} = \{obs_1, obs_2\}$ and for all $n \geqslant 2$, $X_{n,\Phi,\mathtt{ask}} = \{obs_1, obs_2\}$. Proposition 3 gives $Prog(\Phi, \mathtt{ask}) = \mathbf{P_1}(\mathbf{B_1}r \otimes \mathbf{B_1}r) \wedge \mathbf{P}_\infty((\mathbf{B_1}r \otimes \mathbf{B_1}r) \vee (\mathbf{B_1}r \otimes \mathbf{B_1}\neg r)) \equiv \mathbf{P_1}\mathbf{B_2}r \wedge \mathbf{P}_\infty(\mathbf{B_2}r \vee \mathbf{B}_\infty\top) \equiv \mathbf{P_1}\mathbf{B_2}r$.*

The characterization for physical actions is much easier.

Proposition 4 *For any physical action α and any PC $\mathcal{L}_{PS}$ formula Φ, $Prog(\Phi, \alpha) \equiv \mathbf{P}_\infty G(\kappa_\Phi \diamond \alpha)$.*

Moreover, $G(\kappa_\Phi \diamond \alpha)$ can be computed efficiently using Proposition 2 of [Laverny and Lang, 2004].

Lastly, the progression of a positive conjunctive $\mathcal{L}_{PS}$ formula Φ by a BBP π is the canonical $\mathcal{L}^2_{PS}$ formula $Prog(\Phi, \pi)$ defined inductively as follows:

- $Prog(\Phi, \lambda) = \mathbf{P}_\infty\Phi$;
- $Prog(\Phi, \alpha)$ is defined at Definition 4 if α is an action;
- $Prog(\Phi, \mathtt{if}\ \Psi\ \mathtt{then}\ \pi_1\ \mathtt{else}\ \pi_2)$ $= \begin{cases} Prog(\Phi, \pi_1) & \text{if } \Phi \models \Psi \\ Prog(\Phi, \pi_2) & \text{otherwise}\end{cases}$
- $Prog(\Phi, \pi_1; \pi_2) =$ $$\begin{aligned}\bigwedge_{i=1}^{i=n} \mathbf{P_i}(\bigvee_{u+v=i+1} &[Prog([Prog(\Phi, \pi_1)]_u, \pi_2)]_v)\\ &\wedge \mathbf{P}_\infty([Prog([Prog(\Phi, \pi_1)]_\infty, \pi_2)]_\infty)\end{aligned}$$
- $Prog(\Phi, \mathtt{while}\ \Psi\ \mathtt{then}\ \pi') =$ $\begin{cases} Prog(\Phi, \pi'; \pi) & \text{if } \Phi \models \Psi \\ Prog(\Phi, \lambda) & \text{otherwise}\end{cases}$

where

1. for any canonical $\mathcal{L}^2_{PS}$ formula Θ, $[\Theta]_i$ is the strongest positive $\mathcal{L}_{PS}$ formula Ψ (unique up to logical equivalence) such that $\Theta \models \mathbf{P_i}\Psi$.
2. for any positive conjunctive $\mathcal{L}_{PS}$ formulas Φ_1 and Φ_2: $[Prog(\Phi_1 \vee \Phi_2, \pi)]_i = [Prog(\Phi_1, \pi)]_i \vee [Prog(\Phi_2, \pi)]_i$.

Proposition 5 $Prog(\Phi_\kappa, \pi) = \Theta_{\mu(.|\kappa,\pi)}$

Example 5
Let $\pi = (\mathtt{ask}; \mathtt{ask}; \mathtt{if}\ \mathbf{B_2}r \vee \mathbf{B_2}\neg r\ \mathtt{then}\ \lambda\ \mathtt{else}\ \mathtt{ask})$ *and* $\pi' = (\mathtt{while}\ \neg(\mathbf{B_2}r \vee \mathbf{B_2}\neg r)\ \mathtt{do}\ \mathtt{ask})$.

- $Prog(\mathbf{B}_\infty\top, \mathtt{ask}) \equiv \mathbf{P}_\infty(\mathbf{B_1}r \vee \mathbf{B_1}\neg r)$
- $Prog(\mathbf{B}_\infty\top, \mathtt{ask}; \mathtt{ask}) \equiv \mathbf{P_1}(\mathbf{B_2}r \vee \mathbf{B_2}\neg r)$.
- $Prog(\mathbf{B}_\infty\top, \pi) \equiv \mathbf{P_1}(\mathbf{B_2}r \vee \mathbf{B_2}\neg r) \wedge \mathbf{P}_\infty(\mathbf{B_1}r \vee \mathbf{B_1}\neg r)$
- $Prog(\mathbf{B}_\infty\top, \pi') \equiv \mathbf{P}_\infty(\mathbf{B_2}r \vee \mathbf{B_2}\neg r)$

6 Related work

Partially observable Markov decision processes

POMDPs are the dominant approach for acting under partial observability (including nondeterministic actions and unreliable observations). The relative plausibility of observations given states, as well as the notion of progressing a belief state by an action, has its natural counterparts in POMDPs. Now,

[6] Notice that canonical $\mathcal{L}^2_{PS}$ formulas are positive $\mathcal{L}^2_{PS}$ formulas in which there is no disjunction at the level of the $\mathbf{P_i}$ modalities (but disjunctions may appear *in the scope* of a $\mathbf{P_i}$ modality, and of course in the scope of a $\mathbf{B_i}$ modality too).

there are two important differences between POMDPs and our work.

First, in POMDPs policies, branching is conditioned by the observation sequence that has lead to the current belief state; the policy is therefore directly implementable, without the need for an on-line reasoning phase. In our framework, branching conditions are expressed logically, which may allow for much more compact policies than branching on observations. In this view, BBPs can be seen as high-level, compact specifications of POMDP policies (the policy being the implementation of the program). *Our work can thus be seen as a first step towards bridging knowledge-based programs and POMDPs.*

Second, we allow for second-order uncertainty whereas POMDPs get rid of it: if $p_{proj}(p|p_0, \pi)$ is the probability of obtaining the probabilistic belief state p after executing π in p_0, this second-order belief state is systematically reduced into the first-order one $\hat{p}$, following the lottery reduction principle: $\hat{p}(s) = \sum_{p \in PB_S} p_{proj}(p).p(s)$ (where PB_S is the set of probability distributions on S). This is a loss of expressivity, as seen on the following example: consider the action α of tossing a coin and the action β of sensing it. The agent knows that after performing α only, she will be for sure in a belief state where $p(heads) = p(tails) = 0.5$. This differs from projecting the effects of $\alpha; \beta$: then she knows that he will reach either a belief state where she *knows* $heads$ or a belief state where she *knows* $tails$, with equal probability. After reduction, these two plans and their projections can no longer be distinguished – whereas our CBS do distinguish them[7].

Cognitive robotics

Another fairly close area is that of cognitive robotics, especially the work around *Golog* and the situation calculus (e.g., [Reiter, 2001]), which is concerned with logical specifications of actions and programs, including probabilistic extensions and partial observability. [Bacchus *et al.*, 1999] gives an account for the dynamics of probabilistic belief states when perceiving noisy observations and performing physical actions with noisy effectors. [Grosskreutz and Lakemeyer, 2000] considers probabilistic Golog programs with partial observability, with the aim of turning off-line nondeterministic plans into programs that are guaranteed to reach the goal with some given probability. In both works, branching conditions involve objective formulas and there is no account for second-order uncertainty. Bridging belief-based programs and the situation calculus (and Golog) is a promising issue.

Belief revision with unreliable observations

[Boutilier *et al.*, 1998] might be the closest work to ours, as far as the dynamics of belief states in the presence of noisy observations is concerned. Our notion of an observation model (Definition 3) owes a lot to theirs (which also makes use of ranking functions). Now, their objectives depart from ours, as they focus on a general ontology for belief revision and do not consider physical actions nor programs, nor do they give a syntactical way of computing their revision functions.

Counterfactual belief-based programs

[Halpern and Moses, 2004] consider belief-based programs with counterfactuals whose semantics, like ours, is based on ranking functions. They do not allow for graded belief in branching conditions, nor unreliable observations (ranking functions are used for evaluating counterfactuals), but they allow for counterfactual branching conditions, possibly referring to future belief states, such as "if I believe that if I were not sending this message now then my partner might not get this important information, then I should send it". Adding counterfactuals and beliefs about future states to our framework is worth considering for further research.

Acknowledgements

Thanks to the anonymous referees for many relevant and helpful suggestions.

References

[Bacchus *et al.*, 1999] F. Bacchus, J. Halpern, and H. Levesque. Reasoning about noisy sensors and effectors in the situation calculus. *Artificial Intelligence*, 111:171–208, 1999.

[Boutilier *et al.*, 1998] C. Boutilier, N. Friedman, and J. Halpern. Belief revision with unreliable observations. In *Proceedings of AAAI-98*, pages 127–134, 1998.

[Boutilier, 1998] C. Boutilier. A unified model of qualitative belief change: A dynamical systems perspective. *Artificial Intelligence*, 98(1-2):281–316, 1998.

[Fagin *et al.*, 1995] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[Grosskreutz and Lakemeyer, 2000] H. Grosskreutz and G. Lakemeyer. Turning high-level plans into robot programs in uncertain domains. In *Proceedings of ECAI-2000*, pages 548–552, 2000.

[Halpern and Moses, 2004] J. Halpern and Y. Moses. Using counterfactuals in knowledge-based programming. *Distributed Computing*, 17(2):91–106, 2004.

[Laverny and Lang, 2004] N. Laverny and J. Lang. From knowledge-based programs to graded belief based programs, part i: on-line reasoning. In *Proceedings of ECAI-04*, pages 368–372, 2004.

[Levesque and Lakemeyer, 2000] H. Levesque and G. Lakemeyer. *The logic of knowledge bases*. MIT Press, 2000.

[Reiter, 2001] R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[Spohn, 1988] W. Spohn. Ordinal conditional functions: a dynamic theory of epistemic states. In William L. Harper and Brian Skyrms, editors, *Causation in Decision, Belief Change and Statistics*, volume 2, pages 105–134. Kluwer Academic Pub., 1988.

[7] Incidentally, we can reduce a CBS in a similar way: the *reduction* of the CBS μ of the is the *belief state* $\hat{\kappa}_\mu \in B_S$ defined by $\hat{\kappa}_\mu(s) = \min_{\kappa \in B_S} \mu(\kappa) + \kappa(s)$. It is readily checked that $\hat{\kappa}$ is a belief state. Reducing μ into $\hat{\kappa}_\mu$ results in a much simpler (and shorter) structure, which, on the other hand, is much less informative than μ.