NOËL LAVERNY and JÉRÔME LANG

FROM KNOWLEDGE-BASED PROGRAMS TO GRADED BELIEF-BASED PROGRAMS, PART I: ON-LINE REASONING*

ABSTRACT. Knowledge-based programs (KBPs) are a powerful notion for expressing action policies in which branching conditions refer to implicit knowledge and call for a deliberation task at execution time. However, branching conditions in KBPs cannot refer to possibly erroneous beliefs or to graded belief, such as

"`if` my belief that φ holds is high
`then` do some action α
`else` perform some sensing action β".

The purpose of this paper is to build a framework where such programs can be expressed. In this paper we focus on the execution of such a program (a companion paper investigates issues relevant to the off-line evaluation and construction of such programs). We define a simple graded version of doxastic logic KD45 as the basis for the definition of belief-based programs. Then we study the way the agent's belief state is maintained when executing such programs, which calls for revising belief states by observations (possibly unreliable or imprecise) and progressing belief states by physical actions (which may have normal as well as exceptional effects).

1. INTRODUCTION

Knowledge-based programs, or KBPs (e.g. Fagin et al. 1995) are a powerful notion for expressing action policies in which branching conditions refer to implicit knowledge and call for a deliberation task at execution time: informally speaking, branching in KBPs has the following form:

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

where $\mathbf{K}$ is an epistemic (generally 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

`while` I have no strong belief about the direction of the railway station
`do` ask someone

Synthese (2005) 147: 277–321
Knowledge, Rationality & Action 89–133
DOI 10.1007/s11229-005-1350-1

The purpose of this paper is to build a framework for such *belief-based programs* (BBPs). While knowledge states in KBPs are expressed in epistemic logic (usually S5), BBPs need a logic of *graded belief*, where different levels of uncertainty or entrenchment can be expressed. We therefore have to commit to a choice regarding the nature of uncertainty we wish to handle. Rather than reasoning with probabilistic belief states (and therefore introducing probabilistic modalities), which would take us far from usual logics of knowledge or belief such as S5 and KD45,[1] we choose to define belief states as *ordinal conditional functions* (OCF) (Spohn 1988) – also called *kappa-functions*. Introducing OCFs in logic is technically unproblematic (see [Goldszmidt and Pearl 1992; Boutilier et al. 1998, 1999] for logical frameworks of dynamicity and uncertainty based on OCFs); besides, OCFs are expressive enough in many situations where there exists only a small number of "belief degrees"; therefore they are a good trade-off between simplicity and expressivity, as well as between ordinality and cardinality, since they allow for an approximation of probabilities without the technical difficulties raised by the integration of logic and probability. Thus, unsurprisingly, OCFs have been used in several places for building logical frameworks of dynamicity and uncertainty (Goldszmidt and Pearl 1992; Boutilier 1998; Boutilier et al. 1998).

Then, many difficulties arise when considering the way a belief state should be progressed by an action. As in most logical frameworks for reasoning about action we distinguish between *pure sensing actions* who leave the state of the world unchanged and act only on the agent's mental state by giving her some feedback about the actual world, and *purely ontic* (or *physical*) actions, aiming at changing the state of the world without giving any feedback to the agent. This partition can be made without loss of generality (see e.g. [Scherl and Levesque 1993; Herzig et al. 2000), since complex actions (with both ontic effects and feedback) can be sequentially decomposed in two actions, the first being purely ontic and the second one being a pure sensing action.

Let us first consider sensing actions. In S5-based KBPs, observations provided by sensing actions are considered fully reliable; they are taken into account by a pure belief expansion operation. What we need is suitable handling of uncertain initial beliefs, uncertain and partially unreliable observations, and a belief revision operation for incorporating observations into the current belief state. As to ontic actions, BBPs, are intended to cope with the distinction between normal effects and more or less exceptional effects.

We start by defining a graded version of KD45 (Section 2). In Section 3 we show how belief states are *revised* by possibly unreliable observations produced by sensing actions. In Section 4 we show how belief states are *progressed* (or updated) when the agent performs *(physical) actions* which may have alternative effects, some of which being more exceptional than others. Belief-based programs and their relationship to partially observable Markov decision processes are the subject of Section 5. Section 6 discusses further research directions. Since related work pertains to several different areas (depending on whether it relates to graded modalities, revision with uncertain inputs, or progression), we discuss it in the corresponding sections of the document, rather than having a specific Section on related work.

2. $\mathsf{KD45}_G$

2.1. *Graded Beliefs and BBPs*

Our goal being to allow for branching conditions referring to implicit and graded beliefs, we start by generalizing the well-known doxastic logic KD45 so as to allow for *graded belief modalities*.

Let *PS* be a finite set of propositional symbols, The (non-modal) language L_{PS} is defined in the usual way as the propositional language generated from *PS*, the usual connectives, and the Boolean constants $\top$ and $\bot$. Now, we define the language $\mathcal{L}^O_{PS}$ of graded doxastic logic $\mathsf{KD45}_G$.

DEFINITION 1. The language $\mathcal{L}^O_{PS}$ generated from a set of propositional symbols *PS* is defined as follows:

- if φ is an objective formula of L_{PS} then $\mathbf{B}_1\varphi, \mathbf{B}_2\varphi, \ldots, \mathbf{B}_\infty\varphi$ are formulas of $\mathcal{L}^O_{PS}$;
- if φ is an objective formula of L_{PS} then $\mathbf{O}_1\varphi, \mathbf{O}_2\varphi, \ldots, \mathbf{O}_\infty\varphi$ are formulas of $\mathcal{L}^O_{PS}$;
- if Φ and Ψ are formulas of L_{PS} then $\neg\Phi$, $\Phi \vee \Psi$, $\Phi \wedge \Psi$ are formulas of $\mathcal{L}^O_{PS}$.

$\mathbf{B}_i\varphi$, for $i \in \overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$, 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* modality and may be denoted more simply by $\mathbf{K}$ (belief with infinite strength is true knowledge). Modalities $\mathbf{O}_1$, $\mathbf{O}_2$, $\mathbf{O}_n$ and $\mathbf{O}_\infty$ are *only belief* modalities, generalizing *only knowing* (Levesque and Lakemeyer 2000). Intuitively, $\mathbf{O}_{i\varphi}$ means that *all the agent believes to the degree at least i is* φ.

Note that the language $\mathcal{L}^O_{PS}$ considers only *subjective* and *flat* formulas. Neither formulas with nested modalities, nor formulas such as $\varphi \wedge \mathbf{B}_i\psi$, where φ, ψ are both objective, are formulas of $\mathcal{L}^O_{PS}$. This restriction is made for the sake of simplicity; it would be possible to consider a full modal language, and then prove, as it is the case in KD45, that each formula is equivalent to a flat formula, but we leave this technical issue aside since it has little relevance to the issues dealt with in this paper. Likewise, combinations of objective and subjective formulas do not play any role either as far as expressing and interpreting BBPs are concerned. Formulas of $\mathsf{KD45}_G$ are denoted by capital Greek letters Φ, Ψ etc. while objective formulas are denoted by small Greek letters φ, ψ etc.

A BBP is built up from the set of primitive actions *ACT* and usual program constructors. Given a set *ACT* of primitive actions, a BBP is defined inductively as follows:

- the empty plan λ is a BBP;
- for any $\alpha \in ACT$, α is a BBP;
- if π and π' are BBP then $\pi;\pi'$ is a BBP;
- if π and π' are BBP and Φ is a formula of $\mathcal{L}^O_{PS}$, 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* (since formulas of $\mathcal{L}^O_{PS}$ are subjective): 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

$$\begin{aligned}\pi = {} & \texttt{while}\ \neg(\mathbf{B}_2 r \vee \mathbf{B}_2\neg r)\ \texttt{do}\ ask;\\ & \texttt{if}\ \mathbf{B}_2 r\ \texttt{then}\ goright\ \texttt{else}\ goleft\end{aligned}$$

performs the sensing action *ask* until she has a belief firm enough (namely of degree 2) about the way to follow (we'll see in Section 5 that if the *ask* action does not give fully reliable and informative outcomes then this program is not guaranteed to stop).

2.2. *Semantics*

We now give a semantics for interpreting formulas of $\mathcal{L}^{O}_{PS}$. Let $S = 2^{PS}$ be the (finite) set of *states* associated with *PS*. States are denoted by s, s' etc. Rather than writing a state with a subset of *PS*, for sake of clarity, we prefer to write them by listing all propositional symbols with a bar on the symbol when it is false in the state: for instance, if $PS = \{a, b, c, d\}$, then instead of $s = \{b, d\}$ we write $s = \bar{a}b\bar{c}d$; instead of $s = \emptyset$ we write $s = \bar{a}\bar{b}\bar{c}\bar{d}$ etc. If φ is objective then we note $\text{Mod}(\varphi) = \{s \in S | s \models \varphi\}$. For $A \subseteq S$, $\text{Form}(A)$ is the objective formula (unique up to logical equivalence) such that $\text{Mod}(\text{Form}(A)) = A$. If $A = \{s\}$ then we write $\text{Form}(s)$ instead of $\text{Form}(\{s\})$.

DEFINITION 2 (Belief states). An OCF (Spohn 1988), also called a belief state, is a function $\kappa : S \mapsto \overline{\mathbb{N}}$ such that $\min_{s \in S} \kappa(s) = 0$, κ is extended from states to objective formulas by $\kappa(\varphi) = \min\{\kappa(s) | s \models \varphi\}$.

Intuitively, $\kappa(s)$ is the *exceptionality degree* of s, $\kappa(s)$ is usually interpreted in terms of infinitesimal probabilities; $\kappa(s) = k < +\infty$ is then understood as $\text{prob}(s) = o(\varepsilon^k)$, where ε is infinitely small. In particular:

- $\kappa(s) = 0$ means that s is a *normal state* (a normal state is not exceptional, to any degree).
- $\kappa(s) = 1$ means that s is "simply exceptional";
- $\kappa(s) = 2$ means that s is "doubly exceptional';
- $\kappa(s) = +\infty$ means that s is truly impossible. Any state s such that $\kappa(s) < \infty$ is called *a possible state*.

The *normalization constraint* $\min_{s \in S} \kappa(s) = 0$ imposes that there exists at least one normal state. The *void belief state* κ_{void} is defined by $\kappa_{\text{void}}(s) = 0$ for all s.

We now define satisfaction of a $\mathcal{L}^{O}_{PS}$ formula by a belief state.

DEFINITION 3. A model for $\mathsf{KD45}_G$ is simply a an OCF κ. The satisfaction of a formula of $\mathcal{L}_{PS}$ in a model κ is defined by:

- for φ objective and $i \in \overline{\mathbb{N}}, \kappa \models \mathbf{B}_i\varphi$ iff $\kappa(\neg\varphi) \geqslant i$;
- for φ objective and $i \in \overline{\mathbb{N}}, \kappa \models \mathbf{O}_i\varphi$ iff $\forall s \in S,\ \ s \models \neg\varphi \Leftrightarrow \kappa(s) \geqslant i$
- $\kappa \models \Phi \vee \Psi$ iff $\kappa \models \Phi$ or $\kappa \models \Psi$
- $\kappa \models \neg\Phi$ iff $\kappa \not\models \Phi$.

The connectives $\wedge, \rightarrow, \leftrightarrow$ are defined from $\vee$ and $\neg$ in the usual way. Φ is *valid* (resp. *satisfiable*) *iff* it is satisfied in any model (resp, in at least one model). Ψ is a *consequence* of Φ (denoted by $\Phi \models \Psi$) *iff* for any $\kappa, \kappa \models \Phi$ implies $\kappa \models \Psi$. Φ and Ψ are equivalent (denoted by $\Phi \equiv \Psi$) *iff* $\Phi \models \Psi$ and $\Psi \models \Phi$.

Let us briefly comment the definitions.

- $\kappa \models \mathbf{B}_i\varphi$ holds as soon as any model of $\neg\varphi$ is exceptional at least to the degree i (i.e., is such that $\kappa(s) \geqslant i$), or, equivalently, all states such that $\kappa(s) < i$ (i.e., at most i – 1-exceptional) satisfy 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.
- $\kappa \models \mathbf{O}_i\varphi$ holds in κ as soon as the states exceptional at least to the degree i are *exactly* the countermodels of φ, or equivalently, the states exceptional at most to degree $i-1$ are *exactly* the models of φ. In particular, $\mathbf{O}_1\varphi$ is satisfied when all normal states satisfy φ, and all models of φ are normal.

Importantly, $\mathbf{O}_1\top$, means that the agent does not believe anything to the degree 1, therefore nothing either to the degree 2, etc. The only κ satisfying $\mathbf{O}_1\top$ is κ_{void}.

It can be shown easily that each $\mathbf{B}_i$ is a KD45 modality restricted to flat formulas:

PROPOSITION 1. For all φ, ψ in L_{PS} and all i, the following formulas are valid in $\mathsf{KD45}_G$:

1. $\mathbf{O}_i\varphi \rightarrow \mathbf{B}_i\varphi$;
2. $\mathbf{B}_j\varphi \rightarrow \mathbf{B}_i\varphi$ whenever $j \geqslant i$;
3. $\mathbf{B}_i(\varphi \wedge \psi) \leftrightarrow \mathbf{B}_i\varphi \wedge \mathbf{B}_i\psi$;
4. $\neg\mathbf{B}_i\bot$.

Proof.

1. Let κ such that $\kappa \models \mathbf{O}_i\varphi$, which, by definition of the satisfaction relation, is equivalent to $\forall s \in S, s \models \neg\varphi$ *iff* $\kappa(s) \geqslant i$. This implies $\min\{\kappa(s) |\ s \models \neg\varphi\} \geqslant i$, that is, $\kappa(\neg\varphi) \geqslant i$, therefore $\kappa \models \mathbf{B}_i\varphi$.
2. Assume $j \geqslant i$. $\kappa \models \mathbf{B}_j\varphi$ is equivalent to $\kappa(\neg\varphi) \geqslant j$, which implies $\kappa(\neg\varphi) \geqslant i$, i.e., $\kappa \models \mathbf{B}_i\varphi$.

3. $\kappa \models \mathbf{B}_i(\varphi \wedge \psi)$ is equivalent to $\kappa(\neg(\varphi \wedge \psi)) \geqslant i$. Now, $\kappa(\neg(\varphi \wedge \psi)) = \kappa(\neg\varphi \vee \neg\psi) = \min(\kappa(\neg\varphi), \kappa(\neg\psi))$. Therefore, $\kappa \models \mathbf{B}_i(\varphi \wedge \psi)$ is equivalent to $\min(\kappa(\neg\varphi), \kappa(\neg\psi)) \geqslant i$ i.e., $\kappa(\neg\varphi) \geqslant i$ and $\kappa(\neg\psi) \geqslant i$, which is equivalent to $\kappa \models \mathbf{B}_i\varphi$ and $\kappa \models \mathbf{B}_i\psi$, i.e., $\kappa \models \mathbf{B}_i\varphi \wedge \mathbf{B}_i\psi$.
4. Let κ a belief state. Since there exists a s such that $\kappa(s) = 0$, we get $\kappa(\top) = 0$, hence for all $i \geqslant 1$, $\kappa \models \neg\mathbf{B}_i \bot$. □

Remark that due to (3), $\mathbf{B}_i\varphi \rightarrow \mathbf{B}_i\psi$ is valid whenever $\varphi \models \psi$. Remark also that (2) and (3) fail to be valid if we replace $\mathbf{B}_i$ by $\mathbf{O}_i$.

EXAMPLE 1. Let κ defined by $\kappa(a\overline{b}) = 0$, $\kappa(ab) = 1$, $\kappa(\overline{a}b) = 1$ and $\kappa(\overline{a}\overline{b}) = \infty$. Then

- $\kappa \models \mathbf{B}_1 a \wedge \neg\mathbf{B}_2 a$: the agent believes a to the degree 1 (because the (single) normal state, i.e, $a\overline{b}$, satisfies a), but this belief is no firmer than that: a is not believed to the degree 2, because there is a $\neg a$-state s such that $\kappa(s) = 1$, namely $\overline{a}b$.
- $\kappa \models \mathbf{K}(a \vee b)$, because all possible states (namely, $a\overline{b}, ab$ and $\overline{a}b$) satisfy $a \vee b$;
- $\kappa \models \neg\mathbf{B}_1 b$, because the normal state $a\overline{b}$ does not satisfies b.
- $\kappa \models \mathbf{O}_1(a \wedge \neg b)$, because $a \wedge \neg b$ is all the agent believes in the normal states;
- $\kappa \models \mathbf{O}_\infty(a \vee b)$.

The meaning of $\kappa \models \mathbf{O}_i\varphi$ is better understood by the following simple result:

PROPOSITION 2. The two following statements are equivalent:

1. $\kappa \models \mathbf{O}_i\varphi$
2. for every objective formula ψ, $\kappa \models \mathbf{B}_i\psi$ iff $\varphi \models \psi$.

Proof.

(1) $\Rightarrow$ (2) Let $\kappa \models \mathbf{O}_i\varphi$.
(a) Let ψ such that $\varphi \models \psi$. By Proposition 1, $\kappa \models \mathbf{O}_i\varphi$ implies $\kappa \models \mathbf{B}_i\varphi$, therefore, by Proposition 1, $\kappa \models \mathbf{B}_i\psi$.
(b) Let ψ such that $\varphi \not\models \psi$, which entails that there exists a state s such that $s \models \varphi \wedge \neg\psi$. Now, $s \models \varphi$ and $\kappa \models \mathbf{O}_i\varphi$ together imply $\kappa(s) < i$, which in turn implies $\kappa(\neg\psi) < i$ and therefore $\kappa \not\models \mathbf{B}_i\psi$.

(2) $\Rightarrow$ (1) Assume (1) false, i.e., $\kappa \models \neg \mathbf{O}_i\varphi$; then either (c) there is an s such that $s \models \neg\varphi$ and $\kappa(s) < i$, or (d) there is an s such that $s \models \varphi$ and $\kappa(s) \geqslant i$. If (c) holds, then $\kappa \not\models \mathbf{B}_i\varphi$ and then taking $\psi = \varphi$ falsifies (2). If (d) holds, then take $\psi = \neg\text{Form}(s)$. We have $\kappa(\psi) = \kappa(s) \geqslant i$, and yet $\varphi \not\models \psi$, which falsifies(2). □

Syntactically, since the number of states is finite, $\mathbf{O}_i\varphi$ can be defined from the $\mathbf{B}_i$ modalities by the following formula (which is finite only when PS is finite):

PROPOSITION 3.

$$\mathbf{O}_i\varphi \equiv \mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s))$$

Proof.

- We start by showing $\mathbf{O}_i\varphi \models \mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s))$. Let κ such that $\kappa \models \mathbf{O}_i\varphi$, which, by definition of the satisfaction relation, is equivalent to (a) $\forall s \models \varphi$, $\kappa(s) < i$ and (b) $\forall s \models \neg\varphi$, $\kappa(s) \geqslant i$. From point 1 of Proposition 1 we have $\kappa \models \mathbf{B}_i\varphi$. Now, let $s \models \varphi$, which by (a) implies $\kappa(s) < i$. $\kappa(s) < i$, together with $s \not\models \varphi \wedge \neg\text{Form}(s)$, imply $\kappa(\neg(\varphi \wedge \neg\text{Form}(s))) < i$, therefore $\kappa \models \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s))$. This being true for all $s \models \varphi$, and the set of states being finite, we get (d) $\kappa \models \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s))$. From (c) and (d) we get $\kappa \models \mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s))$.
- Now, we show $\mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s)) \models \mathbf{O}_i\varphi$. Let $\kappa \models \neg\mathbf{O}_i\varphi$. Then, either (e) there is a state s such that $s \models \neg\varphi$ and $\kappa(s) < i$ or (f) there is a state s such that $s \models \neg\varphi$ and $\kappa(s) \geqslant i$. If (e) holds, then $\kappa(\varphi) < i$ and therefore $\kappa \models \neg\mathbf{B}_i\varphi$ and *a fortiori* $\kappa \models \neg(\mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s)))$. If (f) holds, then for this state s it holds $\kappa(\varphi \wedge \neg\text{Form}(s)) \geqslant i$, therefore $\kappa \models \mathbf{B}_i(\varphi \wedge \neg\text{Form}(s))$, which entails that $\kappa \models \neg(\bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s)))$. In both cases (e) and (f) we have $\kappa \models \neg(\mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s)))$. This being true for all $\kappa \models \neg\mathbf{O}_i\varphi$, we have $\neg\mathbf{O}_i\varphi \models \neg(\mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s)))$, which is equivalent to $\mathbf{B}_i\varphi \wedge \bigwedge_{s\models\varphi} \neg\mathbf{B}_i(\varphi \wedge \neg\text{Form}(s)) \models \mathbf{O}_i\varphi$. □

EXAMPLE 2. Let $\text{PS} = \{x, y\}$; we have $\mathbf{O}_2(x \vee y) \equiv \mathbf{B}_2(x \vee y) \wedge \neg\mathbf{B}_2 x \wedge \neg\mathbf{B}_2 y \wedge \neg\mathbf{B}_2(x \wedge \neg y \vee \neg x \wedge y)$. The formula $\mathbf{O}_1 x \wedge \mathbf{O}_2 x \wedge \mathbf{O}_3\top$ means that the agent believes only x to the degree 2, that he does not believe more to the degree 1 and that he does not believe anything to a degree > 2.

2.3. *Normal Forms*

We now introduce some useful syntactical notions. A formula of $\mathcal{L}_{PS}^{O}$ is

- a *doxastic atom iff* it is a formula $\mathbf{B}_i\varphi$ where φ is objective.
- *a O-doxastic atom iff* it is a formula $\mathbf{O}_i\varphi$ where φ is objective.
- a *normal positive doxastic* (NPD) *formula iff* Φ is of the form $\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B}_n\varphi_n \wedge \cdots \wedge \mathbf{B}_1\varphi_1$, where $\varphi_\infty, \varphi_1, \ldots, \varphi_n$ are objective formulas such that for all j and $i > j$ we have $\models \varphi_j \rightarrow \varphi_i$.
- a *normal O* (NO) *formula iff* it is of the form $\mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$, where $\varphi_\infty, \varphi_1, \ldots, \varphi_n$ are objective formulas such that for all j and $i > j$ we have $\models \varphi_j \rightarrow \varphi_i$.

EXAMPLE 3.

- $\mathbf{B}_3\neg x$, $\mathbf{K}(\neg x \vee \neg y)$ are doxastic atoms;
- $\mathbf{O}_3\neg x$ is a O-doxastic atom;
- $\mathbf{K}\top \wedge \mathbf{B}_4\top \wedge \mathbf{B}_3 a \wedge \mathbf{B}_2 a \wedge \mathbf{B}_1(a \wedge b)$ is a NPD formula;
- $\mathbf{O}_\infty\top \wedge \mathbf{O}_4\top \wedge \mathbf{O}_3 a \wedge \mathbf{O}_2 a \wedge \mathbf{O}_1(a \wedge b)$ is a NO formula.

When writing a normal positive doxastic formula $\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B}_n\varphi_n \wedge \cdots \wedge \mathbf{B}_1\varphi_1$, we omit subformulas $\mathbf{B}_i\varphi_i$ such that $\varphi_{i+1} \equiv \varphi_i$, as well as tautological subformulas of the form $\mathbf{B}_i\top$: for instance,

$$\mathbf{B}_\infty\top \wedge \cdots \wedge \mathbf{B}_4\top \wedge \mathbf{B}_3 a \wedge \mathbf{B}_2 a \wedge \mathbf{B}_1(a \wedge b)$$

is simply denoted by its equivalent simplified form

$$\mathbf{B}_3 a \wedge \mathbf{B}_1(a \wedge b)$$

Henceforth, formulas such as $\mathbf{B}_2 a, \mathbf{B}_\infty\neg a \wedge \mathbf{B}_1(b \wedge \neg a)$ are considered as normal positive doxastic formulas. The limit case where all φ_i are $\top$ is simply denoted by $\top$ – which is therefore a NPD formula as well. Likewise, $\bot$ is also a NO formula.

Since $\mathbf{B}_i(\varphi \wedge \psi) \leftrightarrow \mathbf{B}_i\varphi \wedge \mathbf{B}_i\psi$ and $\mathbf{B}_i\varphi \rightarrow \mathbf{B}_j\varphi\ (i \geqslant j)$ are valid in $\mathsf{KD45}_G$, any conjunction of doxastic atoms can be equivalently rewritten in NPD form. For instance,

$$\mathbf{B}_3 a \wedge \mathbf{B}_1(a \rightarrow b) \wedge \mathbf{B}_1 c$$

is equivalent to $\mathbf{B}_3 a \wedge \mathbf{B}_1(a \wedge b \wedge c)$.

We also make use of the following syntactical shortcut: for any NPD formula $\Phi = \mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B}_n\varphi_n \wedge \cdots \wedge \mathbf{B}_1\varphi_1$, $\mathbf{Only}(\Phi)$ is the formula $\mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$. Such formulas completely express the agent's belief state; they are satisfied by a single OCF, namely $\kappa_\Phi = G(\Phi)$ defined in Section 2. For instance,

$$\begin{aligned}&\mathbf{Only}(\mathbf{B}_3 a \wedge \mathbf{B}_1(a \wedge b))\\ &\quad = \mathbf{O}_\infty\top \wedge \cdots \wedge \mathbf{O}_4\top \wedge \mathbf{O}_3 a \wedge \mathbf{O}_2 a \wedge \mathbf{O}_1(a \wedge b)\end{aligned}$$

Any belief state κ corresponds to a NO formula Φ_κ, unique up to logical equivalence:

DEFINITION 4 (From belief states to NO formulas and vice versa).

1. for any belief structure κ, $H(\kappa) = \Phi_\kappa$ is the NO formula (unique up to logical equivalence) defined by

   $$\Phi_\kappa = \mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$$

   where

   - $n = \max\{\kappa(s) \mid s \in S \text{ and } \kappa(s) < \infty\}$
   - for all $i \in \{1, \ldots, n, \infty\}$, $\varphi_i =$ Form$(\{s \in S \mid \kappa(s) < i\})$.

2. given a NO formula $\Phi = \mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$, $G(\Phi) = \kappa_\Phi$ is the OCF defined by

   $$\kappa_\Phi(s) = \begin{cases} 0 & \text{if } s \models \varphi_1 \\ i & \text{if } s \models \varphi_{i+1} \wedge \neg\varphi_i \text{ and } i = 1, \ldots, n-1 \\ n & \text{if } s \models \varphi_\infty \wedge \neg\varphi_n \\ +\infty & \text{if } s \not\models \varphi_\infty \end{cases}$$

EXAMPLE 4. Let κ defined by $\kappa([a, \neg b]) = 0$, $\kappa([a, b]) = 1$, $\kappa([\neg a, b]) = 1$ and $\kappa([\neg a, \neg b]) = \infty$. Then

$$H(\kappa) = \mathbf{O}_\infty(a \vee b) \wedge \mathbf{O}_2(a \vee b) \wedge \mathbf{O}_1(a \wedge \neg b)$$

The following property tells that there is a one-to-one correspondence between OCFs and equivalence classes (w.r.t. equivalence on $\mathsf{KD45}_G$) of NO formulas:

PROPOSITION 4. For any NO formula $\Phi = \mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$, $\kappa \models \Phi$ *iff* $\kappa = \kappa_\Phi$.

Proof. Let $\Phi = \mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$.

$\Rightarrow$ Suppose $\kappa \models \Phi$, which is equivalent to the following condition: for all $s \in S$ and every i, $s \models \varphi_i$ *iff* $\kappa(s) < i$. This helps us remarking that (a) for all $s \in S$, $\kappa(s) > n$ implies $\kappa(s) = +\infty$. Consider now the following four cases:

- $\kappa(s) = 0$. In this case, $s \models \varphi_1$ and by definition of κ_Φ, we have $\kappa_\Phi(s) = 0$;
- $\kappa(s) = i$ where $1 \leqslant i \leqslant n-1$. In this case, $s \models \neg\varphi_i \wedge \varphi_{i+1}$ and by definition of κ_Φ, we have $\kappa_\Phi(s) = i$.
- $\kappa(s) = n$. In this case, $s \models \varphi_\infty \wedge \neg\varphi_n$, and by definition of κ_Φ, we have $\kappa_\Phi(s) = n$.
- $\kappa(s) = +\infty$. In this case, $s \models \neg\varphi_\infty$ and by definition of κ_Φ, we have $\kappa_\Phi(s) = \infty$.

Due to (a), these cover all possible cases, therefore $\kappa_\Phi = \kappa$.

$\Leftarrow$ We have to verify that $\kappa_\Phi \models \Phi$. First, we check that for all $s \in S, s \models \neg\varphi_\infty$ *iff* $\kappa_\Phi(s) = +\infty$, therefore $\kappa_\Phi \models \mathbf{O}_\infty\varphi_\infty$. Next, for all $s \in S$ and all $i \leqslant n$, $s \models \neg\varphi_i$ iff $\kappa_\Phi(s) \geqslant i$, therefore $\kappa_\Phi \models \mathbf{O}_i\varphi_i$. Hence, $\kappa_\Phi \models \Phi$. □

COROLLARY 1. $\kappa_{\Phi_\kappa} = \kappa$ and $\Phi_{\kappa_\Phi} \equiv \Phi$.

Proof. Let κ be a belief state. It is easily checked that $\kappa \models \Phi_\kappa$. Now, letting $\Phi = \Phi_\kappa$ in Proposition 4 gives $\kappa \models \Phi_\kappa$ *iff* $\kappa = \kappa_{\Phi_\kappa}$, hence $\kappa_{\Phi_\kappa} = \kappa$. This shows that $H = G^{-1}$ (where NO formulas are identified, by a slight abuse of notation, with their equivalence class w.r.t. logical equivalence), therefore $\Phi_{\kappa_\Phi} \equiv \Phi$. □

Notice that when writing $\Phi_\kappa = \mathbf{O}_\infty\varphi_\infty \wedge \mathbf{O}_{n+1}\varphi_\infty \wedge \mathbf{O}_n\varphi_n \wedge \cdots \wedge \mathbf{O}_1\varphi_1$, φ_i is the formula expressing all the agent believes to the degree i in the belief state κ.

2.4. *Related Work on Modal Logics of Graded Belief*

Although it is original, the construction given in this Section is not the primary goal of the paper. It is very similar to the work on stratified belief bases and possibilistic logic (e.g. (Dubois et al. 1994)) where the duality between (semantical) belief states and (syntactical) NPD formulas can be expressed as well. A multimodal system (with no account for only believing) for possibilistic logic is given in Fariñas del Cerro and Herzig (1991). As for gradual doxastic logics, van der Hock and ch Meyer 1991 define a gradual version of KD45 as well. The interpretation of graded belief is, however, totally different from ours, since $\mathbf{B}_n\varphi$ expresses that φ is true in all worlds except n or less.

3. OBSERVATIONS AND REVISION

3.1. *Combination of Belief States*

We now define the *combination* of belief states, and by isomorphism, the combination of NO formulas. Calling it a "connective" is an abuse of language, since it only connects NO formulas and is therefore not a full-fledged connective.

DEFINITION 5 (OCF combination). Let κ_1 and κ_2 be two OCFs. If $\min_S(\kappa_1+\kappa_2)=\infty$, then $\kappa_1\oplus\kappa_2$ is undefined; otherwise, $\kappa\oplus\kappa_2$ is defined by

$$\forall s\in S,\quad (\kappa_1\oplus\kappa_2)(s)=\kappa_1(s)+\kappa_2(s)-\min_S(\kappa_1+\kappa_2)$$

When defined, we have $\min_S(\kappa_1\oplus\kappa_2)=0$, therefore $\kappa_1\oplus\kappa_2$ is an OCF.

In the particular case of κ_φ defined by

$$\kappa_\varphi(s)=\begin{cases}0 & \text{if } s\models\varphi\\ +\infty & \text{if } s\models\neg\varphi\end{cases}$$

then

$$(\kappa\oplus\kappa_\varphi)(s)=\begin{cases}\kappa(s)-\kappa(\varphi) & \text{if } s\models\varphi\\ +\infty & \text{if } s\models\neg\varphi\end{cases}$$

provided that $\kappa(\varphi)<\infty$. Therefore, $(\kappa\oplus\kappa_\varphi)(s)=\kappa(s|\varphi)$, where $\kappa(.|\varphi)$ is Spohn's conditioning (Spohn 1988).

The intuitive idea behind OCF combination is first illustrated when $\min_S(\kappa_1+\kappa_2)$. When combining the beliefs coming from the sources 1 and 2 (corresponding respectively to κ_1 and κ_2), the combined exceptionality degree of a state s is the sum of the exceptionality of s according to 1 and of that according to 2.

EXAMPLE 5. Consider $\kappa_1 = \kappa_{\Phi_1}$ and $\kappa_2 = \kappa_{\Phi_1}$, where $\Phi_1 = \mathbf{Only}(\mathbf{B}_\infty(a \vee b) \wedge \mathbf{B}_2 a \wedge \mathbf{B}_1(a \wedge b))$ and $\Phi_2 = \mathbf{Only}(\mathbf{B}_1 b)$.

		κ_1	κ_2	$\kappa_1 \oplus \kappa_2$
ab	:	0	0	0
$a\bar{b}$	:	1	1	2
$\bar{a}b$	:	2	0	2
$\bar{a}\bar{b}$	:	∞	1	∞

κ_1 and κ_2 do not conflict: there is a state, namely ab, considered normal by both; hence the identity $\kappa_1 \oplus \kappa_2 = \kappa_1 + \kappa_2$. Now, $(\kappa_1 \oplus \kappa_2)(ab) = 0$ intuitively means that the state ab, considered normal by both κ_1 and κ_2, is considered normal by their combination as well. Next, $a\bar{b}$ being considered simply exceptional by both κ_1 and κ_2, the combination of both considered it doubly exceptional ($(\kappa_1 \oplus \kappa_2)(a\bar{b}) = 2$.) This is justified by the fact that κ_1 and κ_2 are considered as two independent sources: intuitively, if $a\bar{b}$ is the actual state then both sources 1 and 2 have to be wrong. Considering now that source 1 (resp. 2) is wrong about $a\bar{b}$ with probability $o(\varepsilon)$ (because $\kappa_1(a\bar{b}) = \kappa_2(a\bar{b}) = 1$), the probability that both sources are wrong is in $o(\varepsilon^2)$.

When both sources κ_1 and κ_2 conflict, we end up with a $\kappa_1 + \kappa_2$ without any normal state. Renormalizing then just corresponds to making the least exceptional states normal.

EXAMPLE 6. Consider κ_1 as above and $\kappa_3 = \kappa_{\Phi_3}$, where

$$\Phi_3 = \mathbf{Only}(\mathbf{B}_\infty(\neg a \vee \neg b))$$

		κ_1	κ_3	$\kappa_1 + \kappa_3$	$\kappa_1 \oplus \kappa_3$
ab	:	0	∞	∞	∞
$a\bar{b}$	:	1	0	1	0
$\bar{a}b$	:	2	0	2	1
$\bar{a}\bar{b}$	:	∞	0	∞	∞

No state being considered normal by both sources, $a\bar{b}$, being the "closest to normality" when considering both sources, is made normal in their combination.

Up to an isomorphism, $\oplus$ corresponds to the "product combination" of possibility distributions (see Section 3.4. of (Benferhat et al.

2001)), as well as to an infinitesimal version of Dempster's rule of combination (Dempster 1967). The details are in Appendix.

By isomorphism, NO formulas can be combined as well:

DEFINITION 6. For Φ and Ψ two NO formulas we have:

$$\Phi \otimes \Psi = \begin{cases} H(\kappa_\Phi \oplus \kappa_\Psi) = H(G(\Phi) \oplus G(\Psi)) & \text{if defined} \\ \bot & \text{otherwise} \end{cases}$$

Since, due to Corollary 1, there is a one-to-one correspondence between NO formulas (modulo logical equivalence) and belief states, the following holds: let Φ, Ψ are two NO formulas such that $\Phi \otimes \Psi \not\equiv \bot$, then $\kappa \models \Phi \otimes \Psi$ iff $\kappa = \kappa_\Phi \oplus \kappa_\Psi$.

PROPOSITION 5. The following formulas are valid:

1. $\mathbf{Only}(\mathbf{B}_i\varphi) \otimes \mathbf{Only}(\mathbf{B}_j\varphi) \equiv \mathbf{Only}(\mathbf{B}_{i+j}\varphi)$;
2. $\mathbf{Only}(\mathbf{B}_i\varphi) \otimes \mathbf{Only}(\mathbf{B}_j\neg\varphi) \equiv \begin{cases} \mathbf{Only}(\mathbf{B}_{i-j}\varphi) & \text{if } i > j \\ \mathbf{Only}(\mathbf{B}_{j-i}\neg\varphi) & \text{if } j > i \\ \mathbf{Only}(\mathbf{K}\top) & \text{if } i = j \end{cases}$
3. $\Phi \otimes \Psi \equiv \Psi \otimes \Phi$;
4. $\Phi \otimes (\Psi \otimes \Xi) \equiv (\Phi \otimes \Psi) \otimes \Xi)$;
5. $\Phi \otimes \top \equiv \Phi$

Proof.

1. By definition,

$$\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)}(s) = \begin{cases} 0 & \text{if } s \models \varphi \\ i & \text{if } s \not\models \varphi \end{cases}$$

and

$$\kappa_{\mathbf{Only}(\mathbf{B}_j\varphi)}(s) = \begin{cases} 0 & \text{if } s \models \varphi \\ j & \text{if } s \not\models \varphi \end{cases}$$

Therefore

$$(\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)} \oplus \kappa_{\mathbf{Only}(\mathbf{B}_j\varphi)})(s) \begin{cases} 0 & \text{if } s \models \varphi \\ i+j & \text{if } s \not\models \varphi \end{cases}$$

2. We have

$$\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)}(s) = \begin{cases} 0 & \text{if } s \models \varphi \\ i & \text{if } s \not\models \varphi \end{cases}$$

and

$$\kappa_{\mathbf{Only}(\mathbf{B}_j\neg\varphi)}(s) \begin{cases} j & \text{if } s \models \varphi \\ 0 & \text{if } s \not\models \varphi \end{cases}$$

Assume $i > j$. Then $\min(\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)} + \kappa_{\mathbf{Only}(\mathbf{B}_j\neg\varphi)}) = j$; now, if $s \models \varphi$ then $(\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)} + \kappa_{\mathbf{Only}(\mathbf{B}_j\neg\varphi)})(s) = i - j$ and if $s \models \neg\varphi$ then $(\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)} + \kappa_{\mathbf{Only}(\mathbf{B}_j\neg\varphi)})(s) = 0$. The case $j > i$ is symmetric. Lastly, if $i = j$ then $\min(\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)} + \kappa_{\mathbf{Only}(\mathbf{B}_j\neg\varphi)}) = i$, and for every s, $(\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi)} + \kappa_{\mathbf{Only}(\mathbf{B}_j\neg\varphi)})(s) = 0$, hence $\kappa_{\mathbf{Only}(\mathbf{B}_i\varphi\otimes\mathbf{B}_j\neg\varphi)}\kappa_{\text{void}}$.
3. obvious.
4. $((\kappa_1 \oplus \kappa_2) + \kappa_3)(s) = \kappa_1(s) + \kappa_2(s) - \min_S(\kappa_1 + \kappa_2) + \kappa_3(s) - \min_S((\kappa_1 \oplus \kappa_2) + \kappa_3)$. Now, $\min_S((\kappa_1 \oplus \kappa_2) + \kappa_3) = \min_{s\in S}(\kappa_1(s) + \kappa_2(s) - \min_S(\kappa_1 + \kappa_2) + \kappa_3(s)) = \min_{s\in S}(\kappa_1(s) + \kappa_2(s) + \kappa_3(s)) - \min_S(\kappa_1 + \kappa_2)$; therefore, $((\kappa_1 \oplus \kappa_2) \oplus \kappa_3)(s) = \kappa_1(s) \oplus \kappa_2(s) + \kappa_3(s) - \min_S(\kappa_1 + \kappa_2 + \kappa_3)$. This expression is symmetric in κ_1, κ_2 and κ_3, therefore, $(\kappa_1 \oplus \kappa_2) \oplus \kappa_3 = (\kappa_2 \oplus \kappa_3) \oplus \kappa_1)$; by commutativity, we then get $(\kappa_1 \oplus \kappa_2) \oplus \kappa_3 = \kappa_1 \oplus (\kappa_2 \oplus \kappa_3)$. Lastly, by isomorphism we get $\Phi \otimes (\Psi \otimes \Xi) \equiv (\Phi \otimes \Psi) \otimes \Xi)$;
5. obvious from $\kappa_\top = \kappa_{\text{void}}$ and $\kappa \oplus \kappa_{\text{void}} = \kappa$. □

An important corollary of point 1 is that $\Phi \otimes \Phi$ is generally not equivalent to Φ.

As an example, we consider $\Phi_1 \otimes \Phi_2$ where $\Phi_1 = \mathbf{Only}(\mathbf{B}_\infty(a \vee b) \wedge \mathbf{B}_2 a \wedge \mathbf{B}_1(a \wedge b))$ et $\Phi_2 = \mathbf{Only}(\mathbf{B}_1 b)$. We show with the array above that $\Phi_1 \otimes \Phi_2 \equiv \Phi_3$ where $\Phi_3 = \mathbf{Only}(\mathbf{B}_2(a \wedge b)) \wedge \mathbf{B}_\infty$

consists of the truth value of a given objective formula. Unlike most approaches to sensing in reasoning about action and planning, assuming that all sensing actions are basic tests such as in (Scherl and Levesque 1993; van Linder et al. 1994; Levesque 1996; Herzig et al. 2001) becomes a loss of generality when considering belief instead of knowledge: we want to allow for more general sensing actions, whose feedback might be imprecise and/or unreliable.

DEFINITION 7. An Observational believe state, or, for short, an observation, is a belief state κ_{obs}, corresponding to a NO formula $\text{obs} = H(\kappa_{\text{obs}}) = \mathbf{Only}(\mathbf{B}_{\infty}o \wedge \mathbf{B}_{\text{n}}o_n \wedge \cdots \wedge \mathbf{B}_1 o_1)$ (by convention we write $o_{\infty} = o$).

An observation is therefore defined by the belief state it conveys (which, in practice, may be a function of the belief state of the source and the belief that the agent has on the reliability of the source): κ_{obs} is *all we observe* when getting the observation obs. κ_{obs} can also be viewed as the belief state the agent gets into when obtaining obs in the void belief state κ_{void}. The *void observation* obs_{void} is defined by $\text{obs}_{\text{void}} = \mathbf{Only}(\mathbf{K}\top)$ – i.e., $\kappa_{\text{obs}_{\text{void}}} = \kappa_{\text{void}}$.

The outcome of a reliable truth test for a given variable x is an observation of the form $obs \equiv \mathbf{Only}(\mathbf{B}_{\infty}o)$, where $o = x$ or $o = \neg x$. In this case, obs is a *reliable and fully informative observation about* x. If $obs \equiv \mathbf{Only}(\mathbf{B}_{\infty}o)$ where o is a more general formula (such as, for instance, $x \vee y$), then obs is reliable but incomplete; a degenerate case is when $o = \top$: the tautology is observed – obviously with full reliability. Now, a simple observation $\text{obs} \equiv \mathbf{Only}(\mathbf{B}_k o_k)$, where $k < \infty$, is only *partially reliable*. A complex observation is composed of a reliable part (possibly conveying little information, sometimes none at all) and some partially reliable parts – the amount of information obviously decreasing with the reliability level. This rather complex definition is due to the fact that a single observation generally relates to the real state of the world in several ways, with various degrees of uncertainty (exactly as in the Bayesian case). Consider for instance reading the value θ on a temperature sensor, which may for instance correspond to the observation $\text{obs} = \mathbf{Only}(\mathbf{B}_1(t-1 \leqslant \theta \leqslant t+1) \wedge \mathbf{B}_2(t-2 \leqslant \theta \leqslant t+2) \wedge \mathbf{B}_{\infty}(t-5 \leqslant \theta \leqslant t+5))$.

Here is another example. *At 8 in the morning, the agent hears on the radio "due to a strike of a part of the airport staff, today the*

where

- $\psi = \varphi \wedge o$;
- $\forall i \in \mathbb{N}$, $\psi_i = (\varphi_1 \wedge o_i) \vee (\varphi_2 \wedge o_{i-1}) \vee \cdots \vee (\varphi_i \wedge o_1)$;
- $p = \min\{j, \psi_j \not\equiv \bot\}$;
- $m = \max\{j, \psi_{p+j-1} \not\equiv \psi\}$.

Proof. First we show that $\forall i \in \mathbb{N}$, $\mathrm{Mod}(\psi_j) = \{s | \kappa_\Phi(s) + \kappa_{\mathrm{obs}}(s) < j\}$. Let $s \in S$ such that $\kappa_\Phi(s) + \kappa_{\mathrm{obs}}(s) < j$, then $\kappa_\Phi(s) < j - \kappa_{\mathrm{obs}}(s)$, hence $s \models \varphi_{j-\kappa_{\mathrm{obs}}(s)}$ (cf. Definition 4). Furthermore, the same definition implies $s \models o_{\kappa_{\mathrm{obs}}(s)+1}$. Therefore, $s \models \psi_j$. Conversely, let $s \models \psi_j$. Then, by construction of ψ_j, there exist u and v such that $u + v = j + 1$ and $s \models \varphi_u \wedge o_v$. Using definition 4, this implies that $\kappa_\Phi(s) < u$ and $\kappa_{\mathrm{obs}}(s) < v$, *i.e.*, $\kappa_\Phi(s) + \kappa_{\mathrm{obs}}(s) < j$.

This property shows first that $\min_S(\kappa + \kappa_{\mathrm{obs}}) = p - 1$, and then that $\mathrm{Mod}(\psi_{p+i-1}) = \{s | \kappa(s) + \kappa_{\mathrm{obs}}(s) < p + i - 1\}$, *i.e.*, $\mathrm{Mod}(\psi_{p+i-1}) = \{s | \kappa(s) + \kappa_{\mathrm{obs}}(s) - \min_S(\kappa + \kappa_{\mathrm{obs}}) < i\} = \{s | \kappa_\Phi \oplus \kappa_{\mathrm{obs}}(s) < i\}$. Furthermore, it obviously holds that $\mathrm{Mod}(\psi) = \{s \in S | \kappa_\Phi(s) < \infty$ and $\kappa_{\mathrm{obs}}(s) < \infty\} = \{s \in S | (\kappa_\Phi \oplus \kappa_{\mathrm{obs}})(s) < \infty\}$. This shows that $\kappa_{\mathbf{O}_\infty \psi \wedge \mathbf{O}_m \psi_{p+m-1} \wedge \cdots \wedge \mathbf{O}_1 \psi_p} = \kappa_\Phi \oplus \kappa_{\mathrm{obs}}$. Hence, by isomorphism, $\mathbf{O}_\infty \psi \wedge \mathbf{O}_m \psi_{p+m-1} \wedge \cdots \wedge \mathbf{O}_1 \psi_p \equiv \Phi \otimes (\mathbf{O}_\infty o \wedge \mathbf{O}_r o_r \wedge \cdots \wedge \mathbf{O}_1 o_1)$. □

The semantical expression (immediate from Section 3) of the combination of Φ (corresponding to, κ_Φ) by *obs* (corresponding to κ_{obs}) is simply $\kappa(s|\mathrm{obs}) = \kappa(s) + \kappa_{\mathrm{obs}}(s) - \min_S(\kappa + \kappa_{\mathrm{obs}})$, i.e., $\kappa(.|\mathrm{obs}) = \kappa \oplus \kappa_{\mathrm{obs}}$.

Applying Proposition 6 to the specific case of simple observations – of the form obs $=$ $\mathbf{Only}(\mathbf{B}_k o_k)$ – gives a rather long formula that we will not write down here, except in two cases: $k = +\infty$ and $k = 1$. First, when $k = 1$:

COROLLARY 2. Let $\Phi = \mathbf{Only}(\mathbf{B}_\infty \varphi_\infty \wedge \mathbf{B}_n \varphi_n \wedge \cdots \wedge \mathbf{B}_1 \varphi_1)$ and obs $=$ $\mathbf{Only}(\mathbf{B}_1 o_1)$. Then $\Phi \otimes \mathrm{obs} \equiv \mathbf{Only}(\Psi)$ where Ψ is as follows:

Case 1: $\varphi_1 \wedge o_1 \not\equiv \bot$

$$\begin{aligned}\Psi = {} & \mathbf{B}_1(\varphi_1 \wedge o_1) \wedge \mathbf{B}_2(\varphi_1 \vee (\varphi_2 \wedge o_1)) \wedge \cdots \\ & \wedge \mathbf{B}_n(\varphi_{n-1} \vee (\varphi_n \wedge o_1)) \wedge \mathbf{B}_{n+1}(\varphi_n \vee (\varphi_\infty \wedge o_1)) \\ & \wedge \mathbf{B}_\infty \varphi_\infty\end{aligned}$$

Case 2: $\varphi_1 \wedge o_1 \equiv \bot$

$$\begin{aligned}\Psi = {} & \mathbf{B}_1(\varphi_1 \vee (\varphi_2 \wedge o_1)) \wedge \cdots \wedge \mathbf{B}_{n-1}(\varphi_{n-1} \vee (\varphi_n \wedge o_1)) \\ & \wedge \mathbf{B}_n(\varphi_n \vee (\varphi_\infty \wedge o_1)) \wedge \mathbf{B}_\infty \varphi_\infty\end{aligned}$$

Then, when $\text{obs} = \mathbf{Only}(\mathbf{B}_\infty o)$ is a reliable observation, applying Proposition 6 gives

COROLLARY 3. Let $\Phi = \mathbf{Only}(\mathbf{B}_\infty\varphi_\infty \wedge \mathbf{B}_n\varphi_n \wedge \cdots \wedge \mathbf{B}_1\varphi_1)$ and $\text{obs} = \mathbf{Only}(\mathbf{B}_\infty o)$. Assume $\varphi_\infty \wedge o \not\equiv \bot$ and let $p = \min\{j, \varphi_j \wedge o \not\equiv \bot\}$; Then (if $p \leqslant n$)

$$\Phi \otimes \text{obs} \equiv \mathbf{Only}(\mathbf{B}_\infty(\varphi_\infty \wedge o) \\ \wedge \mathbf{B}_{n-p+1}(\varphi_n \wedge o) \wedge \cdots \wedge \mathbf{B}_1(\varphi_p \wedge o))$$

Here is a more intuitive example.

EXAMPLE 7. Consider an agent asking pedestrians about the way to the railway station. Assume there are only two directions, r (right) and $\neg r$ (left). The agent's initial belief state is void ($\kappa_0 = \kappa_{\text{void}}$). When asking a pedestrian, five observations are possible:

- $\mathbf{\Phi}_{\text{obs}_1} = \mathbf{Only}(\mathbf{B}_2 r)$, corresponding to a pedestrian answering "the station is on the right" without hesitation (however, the observation is considered as not fully reliable – since it is known that pedestrians sometimes give wrong indications even when seem to be sure);
- $\mathbf{\Phi}_{\text{obs}_2} = \mathbf{Only}(\mathbf{B}_1 r)$, corresponding to a pedestrian answering "I believe it's on the right but I might be wrong");
- $\mathbf{\Phi}_{\text{obs}_3} = \mathbf{Only}(\mathbf{B}_2 \neg r)$;
- $\mathbf{\Phi}_{\text{obs}_4} = \mathbf{Only}(\mathbf{B}_1 \neg r)$;
- $\mathbf{\Phi}_{\text{obs}_5} = \mathbf{Only}(\mathbf{B}_\infty \top)$ (the pedestrian answers "I have no clue").

We have for instant $\kappa_{\text{obs}_1} = \{(r, 0); (\neg r, 2)\}$ and $\kappa_{\text{obs}_4} = \{(r, 1); (\neg r, 0)\}$. Obviously, $\kappa_0 \oplus \kappa_{\text{obs}_i} = \kappa_{\text{obs}_i}$ for any i.

- After observing obs_2, we have $\kappa_1 = \kappa_0 \oplus \kappa_{\text{obs}_2} = \kappa_{\text{obs}_2}$ and $\Phi_1 = \Phi_0 \otimes \Phi_{\text{obs}_2} = \mathbf{Only}(\mathbf{B}_1 r)$.
- Assume now that the second pedestrian gives obs_2 too, Using Proposition 6, we get:
  - $\psi = \top \wedge \top = \top$;
  - $\psi_1 = r \wedge r = r$;
  - $\psi_2 = (r \wedge \top) \vee (\top \wedge r) = r$;
  - $\psi_3 = (r \wedge \top) \vee (\top \wedge \top) \vee (\top \wedge r) = \top$

  therefore $p = 1$ and $p + m = 3$, hence $\Phi_1 \otimes \Phi_{\text{obs}_1} = \mathbf{Only}(\mathbf{B}_2 r \wedge \mathbf{B}_\infty \top) = \mathbf{Only}(\mathbf{B}_2 r)$.

If the second observation had been obs_4 instead of obs_2 we would have had $\Phi_2 = \Phi_1 \otimes \Phi_{\text{obs}_4} = \mathbf{Only}(\mathbf{B}_\infty \top)$ (the agent comes back to his initial belief state). Indeed,

- $\psi = \top \wedge \top = \top$;
- $\psi_1 = r \wedge \neg r = \bot$;
- $\psi_2 = (r \wedge \top) \vee (\top \wedge \neg r) = \top$

therefore $p = 2$ and $p + m = 2$, hence $\Phi_1 \otimes \Phi_{\text{obs}_2} = \mathbf{Only}(\mathbf{B}_1 \top \wedge \mathbf{B}_\infty \top) = \mathbf{Only}(\mathbf{B}_\infty \top)$.

It can be shown by induction that after p_1 occurrences of obs_1, p_2 of obs_2, p_3 of obs_3, p_4 of obs_4 and p_5 of obs_5 (in any order), iterated combination leads to

- $\mathbf{Only}(\mathbf{B}_q r)$ if $2p_1 + p_2 > 2p_3 + p_4$ and $q = (2p_1 + p_2) - (2p_3 + p_4)$;
- $\mathbf{Only}(\mathbf{B}_q \neg r)$ if $2p_1 + p_2 < 2p_3 + p_4$ and $q = (2p_3 + p_4) - (2p_1 + p_2)$;
- $\mathbf{Only}(\mathbf{B}_\infty, \top)$ if $2p_1 + p_2 = 2p_3 + p_4$.

This example shows how observations *reinforce* prior beliefs when they are consistent with them[3]. It clearly appears that the crucial hypothesis underlying the combination rule is *independence* between the successive observations. Thus, on Example 7, the successive answers are independent (pedestrians do not listen to the answers given by their predecessors). If, on the other hand, we want to express that successive actions are dependent of each other, then we just have to add one or several hidden variables (as commonly done in Markov processes) which would have the effect of blocking (or limiting) the reinforcement[4].

We conclude this section by a discussion on the recent paper (van Ditmarsch 2004), which defines 5 revision operators, four of which appear to be instances of our revision operator:

- *minimal revision* (*1 in (van Ditmarsch 2004)) is the weakest form of revision (in the weak sense, that is, without the so-called "success postulate" telling that when revising by φ, then φ should be believed afterward; it coincides with revision by $\mathbf{Only}(\mathbf{B}_1 \varphi)$– however, the 'eventually successful' property does not hold in our framework because we also consider worlds with an infinite rank, so that if initially all φ-states have an infinite rank, then after any number of revisions by $\mathbf{Only}(\mathbf{B}_1 \varphi)$, the agent still believes $\neg\varphi$.

- *maximal revision* (*2 in (van Ditmarsch 2004)) corresponds to a revision by $\mathbf{Only}(\mathbf{B}_\infty\varphi)$ and therefore to the usual Spohnian revision, used as well in iterated revision frameworks such as (Darwiche and Pearl 1997).
- *"focus on φ" revision* (*4 in (van Ditmarsch 2004)) corresponds to a revision by $\mathbf{Only}(\mathbf{B}_k\varphi)$ such that $k = \max\{\kappa(s)|\kappa(s) < \infty\}$. The effect of such a revision is to make all φ-states that are initially possible more plausible than all $\neg\varphi$-states.
- *"successful minimal"* revision (*5 in (van Ditmarsch 2004)) corresponds to a revision by $\mathbf{Only}(\mathbf{B}_k\varphi)$ with $k = \kappa(\varphi) + 1$ – intuitively, k is the smallest integer such that the revision by $\mathbf{Only}(\mathbf{B}_k\varphi)$ ensures that φ is more believed than $\neg\varphi$.

3.4. *Related Work on Revision by Uncertain Observations*

In addition to (van Ditmarsch 2004) (discussed in Section 3), a close work to ours is (Boutilier et al. 1998), where observational systems allowing for unreliable observations are modeled using OCFs, Their work is less specific than ours (notice that in the absence of ontic actions, our revision process falls in the the class of Markovian observation systems). The main difference between (Boutilier et al. 1998) and our Section 3 is that the revision functions in (Boutilier et al. 1998) remain defined at the semantical level, which, if computed state by state following the definition, needs an exponentially large data structure. Our approach can therefore be viewed as providing a compact representation for a specific class of observation systems. In another line of work, namely (Bacchus et al. 1999), models noisy observations in a probabilistic version of the situation calculus (again, compact representation issues are not considered). (Thielscher 2001) considers noisy sensors as well in a logical framework, but with no graded uncertainty.

Belief transmutations and adjustments (Willams 1994) are based on OCFs too; however, they are based on Spohn's notion of α-conditionalization which, similarly to Jeffrey's rule in probability theory, consist in changing minimally a belief state so as to force a given formula to have the exceptionality degree α; this totally differs from a revision rule enabling an *implicit* reinforcement of belief when the observation is consistent with the initial belief state, as seen in Example 7. Likewise, the work of (Aucher 2004), which defines a logic for public and private announcements with graded plausibility, is based on Spohn's conditionalization as well. The difference between both revision is salient in probability theory as well: Jeffrey's rule has no implicit reinforcing behavior, while Pearls'rule does – see a discussion on both in (Chan and Darwiche 2003). See also (Dubois and Prade 1997) for a panorama of revision rules in numerical formalisms, including OCFs.

4. PROGRESSION

We now consider the case of ontic (or physical) actions. Progressing a belief state by an ontic action is the process consisting of projecting the expected changes implies by the action on the current belief state so as to produce a new belief state, representing the agent's beliefs after the action is performed.

Purely ontic actions may change the state of the world but do not give any feedback. Therefore, given an initial belief state κ and an ontic action α, it is possible to determine the future belief state (after the action is performed) by projecting the possible outcomes of α on the current belief state. This operation is usually called *progression*: $\text{prog}(\kappa, \alpha)$ is the belief state obtained after α is performed in belief state κ. By isomorphism, if Φ is a NO formula, we also define $\text{Prog}(\Phi, \alpha) = H(\text{prog}(G(\Phi), \alpha))$.

4.1. *Semantical Characterization of Progression*

The semantics of progression is defined as in [Boutilier 1998] by means of *OCF transition models*.

DEFINITION 9. An OCF transition model for action α is a collection of OCFs $\{\kappa_{\alpha(.|s), s\in S}\}$.

$\kappa_\alpha(s'|s)$ is the exceptionality degree of the outcome s' when performing action α in state s. Notice that for all $s \in S$, $\min_{s' \in S} \kappa_\alpha(s'|s) = 0$ holds, therefore $\kappa_\alpha(.|s)$ is an OCF. κ_α can be seen as the ordinal counterpart of stochastic transition functions.

DEFINITION 10 (Progression of κ by an ontic action). Given an initial belief state κ and an ontic action α whose dynamics is expressed by the OCF transition model κ_α, the progression of κ by α is the belief state $\kappa' =$ $\text{prog}(\kappa, \alpha)$ defined by

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

This definition appears in several places, including (Goldszmidt and Pearl 1992; Boutilier 1998, Boutilier et al. 1998). It is the ordinal counterpart of $p'(s') = \sum_{s \in S} p(s)p(s'|s,\alpha)$. Notice that κ' is a belief state, because the normalization of *both* κ and $\kappa_\alpha(.|s)$ imply

$$\min_{s \in S}\left\{\min_{s \in S}\{\kappa(s) + \kappa_\alpha(s'|s)\}\right\} = 0$$

i.e., $\min_S \kappa' = 0$

EXAMPLE 8. Consider two blocks A and B lying down on a table; the propositional variable x is true if A is on top of B, false otherwise. A robot can perform the action α consisting in try to put A on B. If A is on B in the initial state, the action has no effect; otherwise, it normally succeeds (*i.e.*, x becomes true), and exceptionally fails (in that case, x remains false). The OCF transition model for α is: $\kappa_\alpha(x|x) = 0$; $\kappa_\alpha(\neg x|x) = \infty$; $\kappa_\alpha(x|\neg x) = 0$; $\kappa_\alpha(\neg x|\neg x) = 1$.

Assume the initial state is κ_{void}, then $\kappa' = \text{prog}(\kappa_{\text{void}}, \alpha) = \{(x,0), (\neg x, 1)\}$; now, $\text{prog}(\kappa', \alpha) = \kappa'' = \{(x,0), (\neg x, 2)\}$. More generally, after performing $\alpha\, n$ times without performing any sensing action (starting from κ_{void}), we get $\text{prog}(\kappa_{\text{void}}, \alpha^n) = \{(x,0), (\neg x, n)\}$, whose associated NO formula is $\mathbf{O_n}x$: after performing action α n times (without sensing), the agent believes to the degree n that A is on B.

Example 8 shows that once again, the underlying hypothesis is the independence between the outcomes of the different occurrences of actions. Indeed, the intuitive explanation of the result of previous example is that after these n executions of α, A is still not on B if and only if all n occurrences of α failed; each of the failures has an exceptionality degree of 1 and failures are independent, henceforth, n successive failures occur with an exceptionality degree of n. Notice that this reinforcement effect is a consequence of the use of $\oplus$ (if conjunction were used instead, we would still get $\mathbf{O}_1 x$ after performing α n times). Again (see Section 3), this reinforcement can be limited or blocked using hidden variables expressing some correlations between the outcomes of the different action occurrences.

In the rest of this section we now show how progression can be computed syntactically, which avoids explicitly computing progression state by state consisting of a straightforward application of the definition.

4.2. *Action Theories with Exceptional Effects*

The first thing we need is a syntactical description of action effects. Therefore, we show that action effects can be described by *graded action theories*, generalizing action theories so as to allow for more or less exceptional action effects.

We first recall briefly that an action theory is a logical theory describing the effects of a given action on a set of variables (or fluents), in a language equipped with a syntactical way of distinguishing between the states of the world *before* and *after* the action is performed. Propositional action theories are usually written by duplicating each variable x of PS in x_t et x_{t+l} (representing x respectively before and after the execution of the action)[5]; this is the way we use for representing graded action theories.

Thus, let $PS_t = \{x_t \mid x \in PS\}$, $PS_{t+1}\{x_{t+1} \mid x \in PS\}$, $S_t = 2^{PS_t}$ and $S_{t+1} = 2^{PS_{t+1}}$. For any formula Φ, let Φ_t (resp. Φ_{t+1}) be the formula obtained from Φ by replacing each occurrence of x by x_t (resp. x_{t+1}). A *graded action theory* is a NO formula of this extended language: $\Sigma_\alpha = \mathbf{Only}(\mathbf{B}_\infty r \wedge \mathbf{B}_n r_n \wedge \cdots \mathbf{B}_1 r_1)$. We just give the graded action theory corresponding to Example 8:

$$\Sigma_\alpha = \mathbf{Only}(\mathbf{B}_\infty(x_t \rightarrow x_{t+1}) \wedge \mathbf{B_1} x_{t+1})$$

The graded action theory can be obtained from a set of causal (dynamic or static) rules through a completion process whose technical details are omitted because they are only little relevant to the subject of this paper. This completion does not present any particular difficulty: it is an easy extension of completion for nondeterministic action theories such as in (Lin 1996; Giunchiglia et al. 2003).

4.3. *Syntactical Characterization of Progression*

We now show how progression can be computed syntactically, which avoids explicitly computing progression state by state consisting of a straightforward application of the definition.

Like for the static case, any OCF transition models correspond to graded action theories and *vice versa*: $\{\kappa_\alpha(.|s), s \in S\}$ induces $\Sigma_\alpha = \mathbf{Only}(\mathbf{B}_\infty r \wedge \mathbf{B}_n r_n \wedge \cdots \wedge \mathbf{B}_1 r_1)$ where $r_i = \text{Form}\{(s'_{t+1}, s_t) \mid \kappa_\alpha(s'_{t+1} \mid s_t) < i\}$.

Now, we recall the definition of *forgetting* a subset of propositional variables X from an objective propositional formula ψ (Lin and Reiter 1994):

1. $\text{forget}(\{x\}, \psi) = \psi_{x\leftarrow\top} \vee \psi_{x\leftarrow\bot}$;
2. $\text{forget}(X \cup \{x\}, \psi) = \text{forget}(\{x\}, \text{forget}(X, \psi))$.

Forgetting is extended to S5 formulas in (Herzig et al. 2003) and is here extended to NO formulas in the following way: if $\Phi = \mathbf{Only}(\mathbf{B}_\infty\varphi \wedge \mathbf{B}_n\varphi_m \wedge \cdots \wedge \mathbf{B}_1\varphi_1)$ and $X \subset \text{Var}(\Phi)$, then $\text{Forget}(X, \Phi) = \mathbf{Only}(\mathbf{B}_\infty\text{forget}(X, \varphi) \wedge \mathbf{B}_n\text{forget}(X, \varphi_n) \wedge \cdots \wedge \mathbf{B}_1\text{forget}(X, \varphi_1))$.

Now we have the following syntactical characterization of progression:

PROPOSITION 7. Let Φ be the NO formula corresponding to the initial belief state κ, and α an ontic action described by an action theory as previously defined. Then

$$\text{Prog}(\Phi, \alpha) \equiv \text{Forget}(PS_t, \Phi_t \otimes \Sigma_\alpha)$$

We start by proving the following Lemma.

LEMMA 1. *Let* $\{X, Y\}$ be a partition of PS and κ an OCF on 2^{PS}. Define $\kappa_X : 2^X \times \overline{\mathbb{N}}$ by: for all $s_X \in 2^X, \kappa_X(s_X) = \min\{\kappa(s_X, s_Y)\ s.t.\ s_Y \in 2^Y\}$. Then $\Phi_{\kappa_X} = \text{Forget}(\Phi_\kappa, Y)$.

Proof. Notice first that $\min \kappa_X = 0$, therefore κ_X is an OCF. Now, let $i \in \{1, \ldots, n, \infty\}$ and $s_X \in 2^X$. Assume $s_X \not\models \text{forget}(Y, \varphi_i)$. Then, by Corollary 5 of Proposition 20 in (Lang et al. 2003), there is no $s_Y \in 2^Y$ such that $(s_X, s_Y) \models \varphi_i$; therefore, for all $s_Y \in 2^Y$ we have $\kappa(s_X, s_Y) \geqslant i$ and $\min_{s_Y \in 2^Y} \kappa(s_X, s_Y) \geqslant i$, i.e., $\kappa_X(s_X) \geqslant i$. Conversely, assume $s_X \models \text{forget}(Y, \varphi_i)$. Then, again from Corollary 5 of Proposition 20 in (Lang et al. 2003), there exists a $s_Y \in 2^Y$ such that $(s_X, s_Y) \models \varphi_i$, therefore $\min_{s_{Y \in 2^Y}} \kappa(s_X, s_Y) < i$, i.e., $\kappa_X(s_X) < i$. In summary, for every i and every $s_X, \kappa_X(s_X) < i$ *iff* $s_X \models \text{forget}(Y, \varphi_i)$, which enables us to conclude that $\Phi_{\kappa_X} = \text{Forget}(\Phi_\kappa, Y)$. □

We now prove Proposition 7.

Proof. We start by defining the cylindrical extension $\tilde{\kappa}$ of κ to $2^{S_t \times S_{t+1}}$ by: for all $s_t \in S_t, \tilde{\kappa}(s_t, s_{t+1}) = \kappa(s_t)$. Then, by definition 10, and using $\min_{(s_t, s_{t+1}) \in S_t \times S_{t+1}}\{\kappa(s_t) + \kappa_\alpha(s_{t+1} \mid s_t)\} = 0$ we get $\kappa'(s_{t+1}) = \min_{s_t \in S_t}\{(\tilde{\kappa} \oplus \kappa_\alpha)(s_t, s_{t+1})\}$. Now, the definition of r_i ($i = 1, \ldots, n, +\infty$) implies $\kappa_\alpha = \kappa_{\Sigma_\alpha}$ and the definition of $\tilde{\kappa}$ implies $\tilde{\kappa} = \kappa_{\Phi_t}$. Therefore, by Definition 5, we get: $\tilde{\kappa} \oplus \kappa_\alpha = G(\Phi_t \otimes \Sigma_\alpha)$. Now, $\kappa'(s_{t+1}) = \min_{s_t \in S_t} \kappa(s_t) + \kappa_\alpha(s_{t+1}|s_t) = \min_{s_t \in S_t}(\tilde{\kappa} \oplus \kappa_\alpha)(s_t, s_{t+1})$. Now, using Lemma 1, $\kappa'(s_{t+1}) = \kappa_{\text{Forget}(\text{Var}_t, \Phi_t \otimes \Sigma_\alpha)}$, which, by isomorphism, is equivalent to $\text{Prog}(\Phi, \alpha) = \text{Forget}(PS_t, \Phi_t \otimes \Sigma_\alpha)$. □

Thus, progression amounts to a combination followed by a forgetting. For the first step, Proposition 6 can be applied again, as shown on the following example. The second step amounts to a sequence of classical forgetting operations.

EXAMPLE 8 (Continued). We have

$$\Sigma_\alpha = \mathbf{Only}(\mathbf{B}_\infty(x_t \to x_{t+1}) \wedge \mathbf{B}_1 x_{t+1})$$

The initial belief state corresponds to

$$\Phi = \mathbf{Only}(\mathbf{B}_1 x)$$

Then,

$$\Phi_t \otimes \Sigma_\alpha = \mathbf{Only}(\mathbf{B}_\infty \psi \wedge \mathbf{B}_n \psi_n \wedge \cdots \wedge \mathbf{B}_1 \psi_1)$$

where

$$\begin{aligned}
\psi &= \top \wedge (x_t \to x_{t+1});\\
\psi_1 &= x_t \wedge x_{t+1};\\
\psi_2 &= (x_t \wedge (x_t \to x_{t+1})) \vee (\top \wedge x_{t+1});\\
\psi_3 &= (x_t \wedge (x_t \to x_{t+1})) \vee (\top \wedge (x_t \to x_{t+1}) \vee (\top \wedge x_{t+1}))
\end{aligned}$$

After simplifying the expression we get $\psi = x_t \to x_{t+1}$; $\psi_1 = x_t \wedge x_{t+1}$; $\psi_2 = x_{t+1}$; $\psi_3 = x_t \to x_{t+1} = \psi$. Next, we get $\Phi_t \otimes \Sigma_\alpha \equiv \mathbf{Only}(\mathbf{B}_\infty(x_t \to x_{t+1}) \wedge \mathbf{B}_1(x_t \wedge x_{t+1}) \wedge \mathbf{B}_2 x_{t+1})$ and Forget $(PS_t, \Phi_t \otimes \Sigma_\alpha) = \mathbf{Only}(\mathbf{B}_\infty \top \wedge \mathbf{B}_1 x_{t+1} \wedge \mathbf{B}_2 x_{t+1}) = \mathbf{Only}\ (\mathbf{B}_2 x_{t+1})$, and finally $\text{Prog}(\Phi, \alpha) = \mathbf{Only}(\mathbf{B}_2 x)$.

Note the importance of combination, which explains the reinforcement obtained when chaining several actions. Such a reinforcement would not be obtained if conjunction were used instead of combination: doing α many times would give $\mathbf{B}_1 x$ again and again.

4.4. *Related Work on Actions with Exceptional Effects*

Goldszmidt and Pearl 1992 and Boutilier 1999 study belief update operators with belief states modeled by OCFs, so as to model exceptional effects of actions. These operators are very similar to our progression for ontic actions from a semantical point of view – but they do not give any syntactical characterization of progression. Shapiro et al. 2000 considers physical and sensing actions in a situation calculus setting, where states are mapped to a plausibility values; these plausibility values are simply inherited from plausibility values in the initial belief state (noisy observations and exceptional effects actions are not considered). See also Baral and Lobo (1997) for a language for describing normal effects in action theories. Lang et al. 2001 define also an update operator for belief states modeled by OCFs, but this operator, which plays more or less for belief update the role played by transmutations for belief revision, is very different from the one given in this article and could not even handle our simple Example 8.

5. ON-LINE EXECUTION OF BBPs

5.1. *Execution and Progression*

BBPs have been defined in Section 2 from a set of propositional symbols PS and a set of primitive actions ACT. For the sake of simplicity, primitive actions are assumed to be either purely physical (or ontic) or purely informative actions: $\text{ACT} = \text{ACT}_P \cup \text{ACT}_I$ (where actions in ACT_P are physical and actions in ACT_I are purely informative, that is, pure sensing actions). This simplification is usual (see Scherl and Levesque 1993; Herzig et al. 2000; Reiter 2001a) and does not induce any loss of generality, as any complex action with both physical and informative effects can be decomposed in two actions performed in sequence, the first one being purely physical and the second one purely informative.

The *on-line execution* of a belief program is a function mapping a pair consisting of an initial belief state and a program to a set of *traces* of the program.

DEFINITION 11 (Traces). A trace is a sequence $\tau = \langle\langle\kappa_t, \alpha_t, \text{obs}_t\rangle_{0\leqslant t\leqslant T-1}, \kappa_T\rangle$ where $T \geqslant 0$ and for all t, κ_t is a belief state, α_t an action and obs_t an observation. (*If* $T = 0$ then $\tau = \langle\kappa_0\rangle$.)

We make use of the following notations:

- if $T \neq 0$ then we write $\tau = \langle\kappa_0, \alpha_0, \text{obs}_0\rangle.\tau'$, where $\tau' = \langle\langle\kappa_t, \alpha_t, \text{obs}_t\rangle_{1\leqslant t\leqslant T-1}, \kappa_T\rangle$.
- $\text{tail}(\tau) = \kappa_T$.

As in Reiter (2001b), each time a program interpreter adds a new action α_t to its action history, the robot (or whatever entity executing the program) also physically performs this action. Since some of these actions are informative actions, we cannot predict off-line the outcome of the program, therefore we must consider a set of possible executions of the program, i.e., a set of traces.

We first have to define an informative action formally. As we said previously, the notion of informative action we need is more complex that actions of the type sense(ψ) used e.g. in Reiter(2001b). These reliable and precise test actions sense(ψ), that send back obs($\mathbf{K}\psi$) if ψ is true in the actual state and obs($\mathbf{K}\neg\psi$) otherwise, are generally assumed to be deterministic, that is, the observation they send back is a function of the actual state of the world. Because we want to allow for possibly unreliable observations, we cannot assume informative actions to be deterministic: the possibility of gathering unreliable pieces of information must come together with the possibility of having several possible observations *even if the state of the world is given*: for instance, in Example 7, given that the station is on the right ($s = r$), we may, for instance, observe either $\mathbf{B}_1 r$, $\mathbf{B}_2 r$, $\mathbf{K}r$, $\top$, $\mathbf{B}_1\neg r$ and $\mathbf{B}_2\neg r$, $\mathbf{K}\neg r$ cannot occur as an observation in that state.

DEFINITION 12 (Feedback function). A feedback function

$$\text{feedback}: \text{ACT} \times S \rightarrow 2^{\text{OBS}}$$

maps each action and each state to a set of observations, satisfying the following requirements:

1. $\text{feedback}(\alpha, s) \neq \emptyset$;
2. if α is ontic then $\text{feedback}(\alpha, s) = \{\text{obs}_{\text{void}}\}$
3. if obs $\in$ feedback(α, s) then obs(s) $< \infty$.

feedback(α, s) is the set of possible observation obtained after performing the sensing action α in state s. Condition 1 requires each action to send back a feedback (possibly void). Ontic actions cannot send any non-void feedback (Condition 2) (alternatively, we could have restricted the definition of the feedback function to informative actions only, but not doing this allows for simpler and shorter definitions further on). Condition 3 ensures a minimum level of consistency between the feedback and the current state, since we exclude that an observation occurs in a state that it totally excludes. The reason for requirement (3) is that revision of κ by obs(**Only**(**K**φ)) is not defined when $\kappa(\varphi) = \infty$ (cf. Section 3); thus, (3) excludes fully contradicting sequences of observations such as obs(**K**φ) followed by obs(**K**$\neg\varphi$) without any ontic action being executed inbetween. In particular, a fully reliable test action such as sense(ψ) as in (Scherl and Levesque 1993; Reiter 2001b) is modeled by the following feedback function:

$$\text{feedback}(\alpha, s) = \begin{cases} \{\text{obs}(\mathbf{Only}(\mathbf{K}\psi))\} & \text{if } s \models \psi \\ \{\text{obs}(\mathbf{Only}(\mathbf{K}\neg\psi))\} & \text{if } s \models \neg\psi \end{cases}$$

But generally, there may be any number of possible outcomes for a given sensing action, including possible void observations ($\text{obs}_{\text{void}} = \mathbf{Only}(\mathbf{K}\top)$).

Now, the agent generally does not know the actual state of the world with precision, which calls for extending the feedback function from states to belief states:

DEFINITION 13 (Subjective feedback). Let feedback be a feedback function. Then the subjective feedback function feedback_S induced by feedback is the function

$$\text{feedback}_S : \text{ACT} \times \text{BS} \rightarrow 2^{\text{OBS}}$$

mapping each action and each belief state to a set of observations, defined by

$$\text{feedback}_S(\alpha, \kappa) = \bigcup \{\text{feedback}(\alpha, s) \mid \kappa(s) < \infty\}$$

It is easily checked that the following properties follow immediately from Definitions 12 and 13.

1. $\text{feedback}_S(\alpha, \kappa) \neq \emptyset$;
2. if α is ontic then $\text{feedback}_S(\alpha, \kappa) = \{\text{obs}_{\text{void}}\}$;
3. if obs $\in \text{feedback}_S(\alpha, \kappa)$ then rev(κ, obs) is defined.

In the specific case where α is a fully reliable truth test that sends back obs(**K**φ) or obs(**K**$\neg\varphi$) then

$$\text{feedback}_S(\alpha, k) = \begin{cases} \{\text{obs}(\mathbf{K}\varphi)\} & \text{if } \kappa \models \mathbf{K}\varphi \\ \{\text{obs}(\mathbf{K}\neg\varphi)\} & \text{if } \kappa \models \mathbf{K}\neg\varphi \\ \{\text{obs}(\mathbf{K}\varphi)\}, \{\text{obs}(\mathbf{K}\neg\varphi)\} & \text{otherwise} \end{cases}$$

In other words, if the agent already *knows* that φ is true, then the only possible feedback is observing that φ is true (otherwise the agent would have had an incorrect infinite belief, that is, incorrect knowledge).

We now define the set of possible executions of a BBP π in an initial belief state κ.

DEFINITION 14 (possible executions of a BBP). A trace $\tau = \langle\langle\kappa_t, \alpha_t, \text{obs}_t\rangle_{0\leqslant t\leqslant T-1}, \kappa_T\rangle$ is a possible execution of the BBP π in the belief state κ iff one of the following conditions is satisfied:

1. $\tau = \langle\kappa\rangle$ and $\pi = \lambda$;
2. (a) $\tau = \langle\kappa, \alpha, \text{obs}\rangle.\tau'$ where $\alpha \in \text{ACT}_P$ (b) $\pi = \alpha; \pi'$, (c) obs $=$ obs_{void} and (d) τ' is a possible execution of ϕ' in prog (κ, α);
3. (a) $\tau = \langle\kappa, \alpha, \text{obs}\rangle.\tau'$ where $\alpha \in \text{ACT}_I$, (b) $\pi = \alpha; \pi'$, (c) obs $\in$ $\text{feedback}_S(\alpha, \kappa)$ and (d) τ' is a possible execution of π' in rev (κ,obs);
4. (a) $\tau = \langle\kappa, \alpha, \text{obs}\rangle.\tau'$, (b) $\pi = \texttt{if}\ \Phi\ \texttt{then}\ \pi_1\ \texttt{else}\ \pi_2; \pi_3$, and (c) either $\kappa \models \Phi$ and τ is a possible execution of $(\pi_1; \pi_3)$ in κ, or $\kappa \models \neg\Phi$ and τ is a possible execution of $(\pi_2; \pi_3)$ in κ.
5. (a) $\tau = \langle\kappa, \alpha, \text{obs}\rangle.\tau'$, (b) $\pi = \texttt{while}\ \Phi\ \texttt{do}\ \pi_1; \pi_2$, and (c) either $\kappa \models \Phi$ and τ is a possible execution of $(\pi_1; \pi)$ in κ, or $\kappa \models \neg\Phi$ and τ is a possible execution of π_2 in κ.

We denote by exec(π, κ) the set of possible executions of π in κ.

It is easily shown that any BBP has at least one possible execution in any belief state. There may be infinitely many such possible executions, as shown in the following example.

EXAMPLE 9. Consider Example 7 again. Here are some possible executions of $\pi = \texttt{while}\ \neg(\mathbf{B}_2 r \vee \mathbf{B}_2\neg r)\ \texttt{do ask}$ in $\kappa_{\mathbf{Only}(\top)}$, where $\kappa_0 = \kappa_{\mathbf{Only}(\top)}$, $\kappa_1 = \kappa_{\mathbf{Only}(\mathbf{B}_{1r})}$, $\kappa_2 = \kappa_{\mathbf{Only}(\mathbf{B}_1\neg r)}$, $\kappa_3 = \kappa_{\mathbf{Only}(\mathbf{B}_2 r)}$, $\kappa_4 = \kappa_{\mathbf{Only}(\mathbf{B}_2\neg r)}$, $\text{obs}_1 = \text{obs}(\mathbf{Only}(\mathbf{B}_1 r))$ and $\text{obs}_2 = \mathbf{Only}(\mathbf{B}_1\neg r)$.

- $\langle\langle\kappa_0, \text{ask}, \text{obs}_1\rangle, \langle\kappa_1, \text{ask}, \text{obs}_1\rangle, \kappa_2\rangle$;
- $\langle\langle\kappa_0, \text{ask}, \text{obs}_2\rangle, \langle\kappa_3, \text{ask}, \text{obs}_1\rangle, \kappa_4\rangle$;
- $\langle\langle\kappa_0, \text{ask}, \text{obs}_1\rangle, \langle\kappa_1, \text{ask}, \text{obs}_2\rangle, \langle\kappa_0, \text{ask}, \text{obs}_1\rangle,$
  $\langle\kappa_1, \text{ask}, \text{obs}_1\rangle, \kappa_2\rangle$;
- $\langle\langle\kappa_0, \text{ask}, \text{obs}_1\rangle, \langle\kappa_1, \text{ask}, \text{obs}_2\rangle, \langle\kappa_0, \text{ask}, \text{obs}_2\rangle,$
  $\langle\kappa_3, \text{ask}, \text{obs}_2\rangle, \kappa_2\rangle$; etc.

There are infinitely many possible executions of π. They can be finitely described by the regular expression $[(ab \cup cd)^*; (ae \cup cf)]$, where $a = \langle \kappa_0, \text{ask}, \text{obs}_1 \rangle$, $b = \langle \kappa_1, \text{ask}, \text{obs}_2 \rangle$, $c = \langle \kappa_0, \text{ask}, \text{obs}_2 \rangle$, $d = \langle \kappa_3, \text{ask}, \text{obs}_1 \rangle$, $e = \langle \kappa_1, \text{ask}, \text{obs}_1 \rangle$ and $f = \langle \kappa_3, \text{ask}, \text{obs}_2 \rangle\rangle)$.

We now show how progression can be extended from single actions to belief-based programs, and then show the correspondance with the set of possible executions of the program.

DEFINITION 15 (Progression of an initial belief state by a BBP). Given a BPP π and an *NO* formula Φ, the progression of Φ by π is the set of *NO* formulas $\text{Prog}(\Phi, \pi)$ defined inductively by

- $\text{Prog}(\Phi, \lambda) = \{\Phi\}$;
- if $\pi = \alpha; \pi'$ with $\alpha \in \text{ACT}_P$ then $\text{Prog}(\Phi, \pi) = \text{Prog}(\text{Prog}(\Phi, \alpha), \pi')$
- if $\pi = \alpha; \pi'$ with $\alpha \in \text{ACT}_I$ then

$$\text{Prog}(\Phi, \pi) = \bigcup_{\text{obs} \in \text{feedback}(\alpha, \kappa_\Phi)} \text{Prog}(\Phi \otimes \text{obs}, \pi')$$

- if $\pi = (\texttt{if}\ \Psi\ \texttt{then}\ \pi_1\ \texttt{else}\ \pi_2); \pi_3$ then

$$\text{Prog}(\Phi, \pi) = \begin{cases} \text{Prog}(\Phi, (\pi_1; \pi_3)) & \text{if } \Phi \models \Psi \\ \text{Prog}(\Phi, (\pi_2; \pi_3)) & \text{otherwise} \end{cases}$$

- if $\pi = (\texttt{while}\ \Psi\ \texttt{do}\ \pi_1); \pi_2$ then

$$\text{Prog}(\Phi, \pi) = \begin{cases} \text{Prog}(\Phi, (\pi_1; \pi)) & \text{if } \Phi \models \Psi \\ \text{Prog}(\Phi; \pi_2) & \text{otherwise} \end{cases}$$

The following result guarantees that the syntactical way of computing progression is correct.

PROPOSITION 8. $\Psi \in \text{Prog}(\pi, \Phi)$ iff there is an execution $\tau \in \text{exec}(\pi, \kappa_\Phi)$ such that $\text{tail}(\tau) = \kappa_\Psi$.
Remark that an equivalent formulation of the above identity is

$$\text{tail}(\text{exec}(\pi, \kappa)) = \{\kappa_\Psi, \Psi \in \text{Prog}(\pi, \Phi_\kappa)\}$$

where $\text{tail}(X) = \{\text{tail}(\tau) \text{ s.t. } \tau \in X\}$.

Proof. By induction on the size of π. We first define the size of a BBP inductively by: size(λ) = 0; size(α) = 1 for $\alpha \neq \lambda$; size($\pi;\pi'$) = size(π)+ size(π'); size(`if` Φ `then` π_1 `else` π_2) = max(size(π_1), size(π_2))+1; size(`while` Φ `do` π') = size(π')+1. Let us consider the induction hypothesis

$$\text{I(m)}: \text{ for all } \pi \text{ such that size}(\pi) \leqslant m \text{ and for all } \kappa,$$
$$\text{tail}(\text{exec}(\pi,\kappa)) = \{\kappa_\Psi \text{ s.t. } \Psi \in \text{Prog}(\Phi_{\kappa,\pi})\}$$

If $\pi = \lambda$ then exec(π, Φ) = $\{\Phi\} = \{\Phi\}_{\kappa_\Phi}$ by Corollary 1. Therefore $I(0)$ is verified. Assume now that $I(m)$ is verified and let π be a BBP such that size(π) = $m+1$.

- if $\pi = \alpha;\pi'$ and $\alpha \in \text{ACT}_P$. Since size(π') = m, $I(m)$ implies that exec(π',κ) = $\{\kappa_\Psi, \Psi \in \text{Prog}(\Phi_\kappa,\pi')\}$. Then we have the following chains of equivalences:
  $\Psi \in \text{Prog}(\Phi,\pi)$
  iff $\Psi \in \text{Prog}(\Phi,(\alpha;\pi'))$
  iff $\Psi \in \text{Prog}(\text{Prog}(\Phi,\alpha)\pi')$
  iff there is a $\tau' \in$ exec($\pi', \kappa_{\text{Prog}(\Phi,\alpha)}$) such that tail($\tau'$) = κ_Ψ
  iff there is a $\tau' \in$ exec(π', Prog(κ_Φ,α)) such that tail(τ') = κ_Ψ
  iff there is a $\tau \in$ exec(($\alpha;\pi'$), κ_Φ) such that tail (τ') = κ_Ψ
  iff there is a $\tau \in$ exec(π,κ_Φ) such that tail (τ) = κ_Ψ.
  iff $\kappa_\Psi \in$ tail(exec(π,κ_Φ)).
- let $\pi = \alpha;\pi'$ and $\alpha \in \text{ACT}_I$. Again, exec(π',κ) = $\{\kappa_\Psi, \Psi \in \text{Prog}(\Phi_\kappa,\pi')\}$ holds the induction hypothesis. Then we have the following chains of equivalences:
  $\Psi \in \text{Prog}(\Phi,\pi)$
  iff $\Psi \in \text{Prog}(\Phi,(\alpha;\pi'))$
  iff $\Psi \in \bigcup\{\text{Prog}(\Phi \otimes \text{obs}) \mid \text{obs} \in \text{feedback}(\alpha,\Phi)\}$
  iff there is a obs $\in$ feedback(α,Φ) and a $\tau \in$ exec ($\pi',\kappa_{\Phi\otimes\text{obs}}$)such that tail($\tau$) = κ_Ψ
  iff there is a obs $\in$ feedback(α,Φ) and a $\tau \in$ exec (π', rev($\kappa_\Phi,\kappa_{\text{obs}}$)) such that tail ($\tau$) = κ_Ψ
  iff there is a $\tau \in$ exec(($\alpha;\pi'$), κ_Φ) such that tail(τ) = κ_Ψ
  iff $\kappa_\Psi \in$ tail(exec(π,κ_Ψ))
- let π = (`if` Γ `then` π_1 `else` π_2); π_3. Note that size($\pi_1;\pi_3$) $\leqslant m$ and size($\pi_2;\pi_3$) $\leqslant m$, therefore the induction hypothesis can be applied to $\pi_1;\pi_3$ and to $\pi_2;\pi_3$. Then we have $\Psi \in$ Prog(Φ,π)

iff either $\Phi \models \Gamma$ and $\Psi \in$ Prog$(\Phi, (\pi_1; \pi_3))$ or $\Phi \not\models \Gamma$ and $\Psi \in$ Prog$(\Phi, (\pi_2; \pi_3))$

iff either $\kappa_\Phi \models \Gamma$ and there is a $\tau \in$ exec $((\pi_1; \pi_3), \kappa_\Phi)$ such that tail$(\tau) = \kappa_\Psi$ or $\kappa_\Phi \not\models \Gamma$ and there is a $\tau \in$ exec$((\pi_2; \pi_3), \kappa_\Phi)$ such that tail$(\tau) = \kappa_\Psi$

iff $\kappa_\Psi \in$ tail(exec(π, κ_Φ)).

- the case π = `while` Γ `do` $\pi_1; \pi_2$ is similar to the latter.

Therefore, in all cases we have $\Psi \in$ Prog(Φ, π) *iff* $\kappa_\Psi \in$ tail(exec(π, κ_Φ)), or equivalently, tail(exec(π, κ)) = $\{\kappa_\Psi$ s.t. $\Psi \in$ Prog$(\Phi_\kappa, \pi)\}$. This being true for any π of size $m+1$, we have shown that the induction hypothesis carries on from m to $m+1$, which completes the proof.

□

EXAMPLE 10. Consider Example 7 again. We have the following:

$$\text{Prog}(\pi, \text{Only}(\top)) = \{\text{Only}(\mathbf{B}_2 r), \text{Only}(\mathbf{B}_2 \neg r)\}$$

Remark here that Prog$(\pi, \mathbf{Only}(\top))$ is finite although exec $(\pi, \kappa_{\mathbf{Only}(\top)})$ is infinite.

5.2. *BBP as Implicit and Compact Representations of POMDP Policies*

POMDPs are the dominant approach for planning under partial observability (including nondeterministic actions and unreliable observations) – see for instance (Kaelbling et al. 1998; Bonet and Geffner 2001) for two of the most relevant references on planning with POMDPs). The relative plausibility of observations given states, as well as the notion of progressing a belief state by an action, has its counterparts in POMDPs. Now, there are two important differences between POMDPs and our work.

A POMDP policy σ is a labeled automaton, that is, a graph whose vertices are labeled by actions and edges by observations, and the outcoming edges from a vertex v labeled by α are labeled by a possible feedback obs of α (in particular, if α is ontic then there is a unique outcoming edge from v, labeled by obs_{void}).

Unlike a BBP, a policy can be followed without needing to perform a deduction task for evaluating a branching condition: a policy is executed just by following the observation flow and executing the indicated actions.

Given a BBP π and an initial belief state, it is possible to "compile" π into a policy σ, by simulating its execution and evaluating the branching conditions for each possible observation sequence. For the sake of simplicity we define this induced policy as a tree; it is then possible to reduce the tree into a smaller graph by a standard automaton minimization process.

In the following definition we denote by Tree(α, $\langle \text{obs}_1, \tau_1 \rangle, \ldots, \langle \text{obs}_p, \tau_p \rangle$) the tree whose root is labelled by α and containing p subtrees $\tau_1, \ldots, \tau_p$, labeled respectively by $\text{obs}_1, \ldots, \text{obs}_p$.

DEFINITION 16. Let π be a BBP and κ a belief state. Then the policy $\sigma =$ policy (π, κ) induced by π and κ is defined inductively by

- policy(λ, κ) is the tree composed of a single vertex labeled by λ;
- if $\pi = \alpha; \pi'$ with $\alpha \in \text{ACT}_P$ then

$$\text{policy}(\pi, \kappa) = \text{Tree}(\alpha, \langle \text{obs}_{\text{void}}, \text{policy}(\pi', \text{prog}(\kappa, \alpha)) \rangle)$$

- if $\pi = \alpha; \pi'$ with $\alpha \in \text{ACT}_I$ then

$$\text{policy}(\pi, \kappa) = \text{Tree}(\alpha, \langle \text{obs}_1, \text{policy}(\pi', \text{rev}(\kappa, \text{obs}_1)) \rangle, \ldots, \langle \text{obs}_p, \text{policy}(\pi', \text{rev}(\kappa, \text{obs}_p)) \rangle)$$

  where $\{\text{obs}_1, \ldots, \text{obs}_p\} = \text{feedback}_S(\alpha, \kappa)$.
- if $\pi = (\texttt{if } \Phi \texttt{ then } \pi_1 \texttt{ else } \pi_2); \pi_3$ then

$$\text{policy}(\pi, \kappa) = \begin{cases} \text{policy}((\pi_1; \pi_3), \kappa) & if\ \kappa \models \Phi \\ \text{policy}((\pi_2; \pi_3), \kappa) & \text{otherwise} \end{cases}$$

- if $\pi = (\texttt{while } \Phi \texttt{ do } \pi'); \pi''$ then

$$\text{policy}(\pi, \kappa) = \begin{cases} \text{policy}((\pi'; \pi), \kappa) & if\ \kappa \models \Phi \\ \text{policy}(\pi'', \kappa) & \text{otherwise} \end{cases}$$

The crucial difference between a BBP and the policy implementing it is in the expression of branching conditions :

- in a BBP branching conditions are *subjective*, since they refer to the current belief state of the agent.
- in a POMDP policy, branching conditions are *objective*: the next action is dictated by the last observation made.

This difference in the nature of branching conditions has two important practical consequences:

1. a policy is directly implementable (at each point in the policy execution, the next action to be performed is specified directly from the feedback and is therefore determined in linear time, just by following the edge corresponding to the observation made in the policy graph). Contrariwise, a BBP is not directly implementable, since branching conditions have first to be evaluated. Evaluating a branching condition is a coNP-hard problem that has to be solved on-line: thus, BBPs need a *deliberation phase* when being executed, while policies do not.
2. a BBP is a much more compact description of the policy than the explicit specification of the policy itself. Indeed, policies induced by BBPs without `while` statements are, in the worst case, exponentially larger than that the BBP they implement.

A policy is a particular case of a *protocol* in the sense of Fagin et al. (1995). A single-agent protocol maps the *local state* of the agent to an action; here, a local state is defined by the sequence of observations and actions performed so far (and thus corresponds to a vertex in the policy tree). A more extensive discussion on the differences between protocols and KBPs can be found in Fagin et al. (1995). We end up this discussion by giving two examples.

EXAMPLE 11. Consider the BBP

$$\text{while}\neg(\mathbf{B}_2 x \vee \mathbf{B}_2 \neg x)\ \text{do ask}$$

applied in the void initial belief state κ_{void}.

Then the policy σ implementing π, as it is defined above, is an infinite tree, which can easily be shown to be reductible to the following finite graph $G = \langle V, E\rangle$ defined by:

- $V = \{v_\top, v_{\mathbf{B}_1 x}, v_{\mathbf{B}_2 x}, v_{\mathbf{B}_1 \neg x}, v_{\mathbf{B}_2 \neg x}\}$;
- $v_\top, v_{\mathbf{B}_1 x}$ and $v_{\mathbf{B}_1 \neg x}$ are labeled by ask whereas $v_{\mathbf{B}_2 x}$ and $v_{\mathbf{B}_2 \neg x}$ are labeled by λ.
- $E = \{(v_\top, \text{obs}(\mathbf{B}_1 x),\ v_{\mathbf{B}_1 x}),\ (v_\top, \text{obs}(\mathbf{B}_1 \neg x),\ v_{\mathbf{B}_1 \neg x}),\ (v_{\mathbf{B}_1 x}, \text{obs}(\mathbf{B}_1 x),\ v_{\mathbf{B}_2 x}), (v_{\mathbf{B}_1 x}, \text{obs}(\mathbf{B}_1 \neg x),\ v_\top), (v_{\mathbf{B}_1 \neg x}, \text{obs}(\mathbf{B}_1 \neg x),\ (v_{\mathbf{B}_2 \neg x})), (v_{\mathbf{B}_1 \neg x}, \text{obs}(\mathbf{B}_1 x), v_\top)\}$, where $(v_\top, \text{obs}(\mathbf{B}_1 x), v_{\mathbf{B}_1 x})$ denotes an edge from $v_\top$ to $v_{\mathbf{B}_1 x}$ labeled by $\text{obs}(\mathbf{B}_1)x)$, etc.

EXAMPLE 12. Consider a model-based diagnosis problem, with n components $1, \ldots, n$. For each component i, the propositional variable ok(i) represents the status of component i (working state if ok(i) is true, or failure state otherwise). Σ is a propositional formula expressing links between the components, given some background knowledge about the system plus possibly some initial measurements: for instance, $\Sigma = (\neg ok(1) \vee \neg ok(2)) \wedge (\neg ok(1) \vee \neg ok(3)) \wedge \neg ok(4)$ means that one of the components 1 and 2 is faulty, one of the components 1 and 3 is, and component 4 is faulty. Each component can be inspected by means of a purely informative action inspect(i) whose feedback is either $\mathbf{K}ok(i)$ or $\mathbf{K}\neg ok(i)$, and repaired by means of an ontic action repair(i) whose effect is ok(i). For the sake of simplicity we assume that beliefs are nongraded. Consider the following BBP π:

```
while ¬K(ok(1) ∧ ··· ∧ ok(n))
do
      pick a i such that ¬Kok(i)
      if K¬ok(i)
      then repair(i)
      else if ¬Kok(i)
           then inspect(i)
           end if
      end if
end while
```

It can be shown that this program is guaranteed to stop after less than $2n$ actions. The size of the policy σ induced by π is, in the worst case, exponential in n.

To sum up, BBPs are a smart and compact way of specifying policies, which, on the other hand, requires much more computational tasks at execution time than the explicit policy.

Our work can thus be seen as a first step towards bridging KBPs and POMDPs. It would be interesting to go further and to build a language for BBPs describing "real" POMDP (with probabilistic belief states). This would require a rather deep modification of our framework, since probabilistic modalities are more complex than our graded belief modalities. This issue of designing probabilistic programs as compact description of POMDP policies is a promising topic that we leave for further research.

5.3. *Detailed Example*

Let us consider a last example, inspired from Levesque (1996).

EXAMPLE 13. The agent has a bowl, initially empty, and a box of 3 eggs; each egg is either good or rotten. There are three actions:

- `takeNewEgg` is a pure ontic action resulting in the agent having in his hand a new egg from the box; since this new egg may be good or rotten, `takeNewEgg` is nondeterministic; however, its normal result is the agent having a good egg in his hand; getting a rotten egg in hand is 1-exceptional.
- `testEgg` is a pure sensing action, consisting of smelling the egg; its feedback contains two possible observations: $\mathbf{Only}(\mathbf{B}_1 \mathrm{g})$ and $\mathbf{Only}(\mathbf{B}_1 \neg \mathrm{g})$. (Note that smelling is here considered as not fully reliable).
- `putIntoBowl` is a pure ontic action consisting in breaking the egg into the bowl; it results in the content of the bowl being spoiled if the egg is rotten, and in the bowl containing one more egg if the egg is good.

This domain can be modeled using the following set of variables:

- `egg` (the agent holds an egg in his hand);
- `g` (the last egg taken from the box is a good one);
- $\mathtt{in}(i)$ for $i \in \{0, \ldots, 3\}$ (the bowl contain exactly i eggs);
- `spoiled` (the bowl contain at least one rotten egg);
- and the derived fluents $\mathtt{om}(i)$, $i = 0, \ldots, 3$, defined from the other fluents by: $\mathtt{om}(0) \equiv \mathtt{in}(0) \vee \mathtt{spoiled}$ and for all $i > 0$, $\mathtt{om}(i) \equiv \mathtt{in}(i) \wedge \neg \mathtt{spoiled}$.

The ontic action `takeNewEgg` is modeled by the following transition system: for any state s, let $s + (\mathrm{egg}, \mathrm{g})$ (resp. $s + (\mathrm{egg}, \neg \mathrm{g})$) the state obtained from s by (a) assigning `g` to true (resp. false) and (b) assigning `egg` to true, Then, for any s, $\kappa(s + (\mathrm{egg}, \mathrm{g}) | s) = 0$ and $\kappa(s + (\mathrm{egg}, \neg g) | s) = 1$. The action theory corresponding to `take-NewEgg` is

$$\Sigma_{\mathtt{takeNewEgg}} = \mathbf{K}\left(\mathtt{egg}_{t+1} \wedge \left(\bigwedge_i \mathtt{in}(i)_{t+1} \leftrightarrow \mathtt{in}(i)_t\right) \wedge (\mathtt{spoiled}_{t+1} \leftrightarrow \mathtt{spoiled}_t)\right) \wedge \mathbf{B}_1 \mathrm{g}_{t+1}$$

The ontic action putIntoBowl is modeled by the following transition system: for any state s, let i_s be the number of eggs in the bowl in s (that is, $s \models \texttt{in}(i_s)$ and $s \models \neg\texttt{in}(j)$ for all $j \neq i(s)$). Let next(putIntoBowl, s) be the state defined by:

- if $s \models \texttt{egg} \wedge g$ then next(putIntoBowl, s) is the state obtained from s by (a) assigning egg to false; (b) assigning $\texttt{in}(i_s)$ to false and $\texttt{in}(i_s+1)$ to true (the rest being unchanged);
- if $s \models \texttt{egg} \wedge \neg\texttt{g}$ then next(putIntoBowl, s) is the state obtained from s by (a) assigning egg to false; (b) assigning $\texttt{in}(i_s)$ to false and $\texttt{in}(i_s+1)$ to true; (c) assigning spoiled to true (the rest being unchanged);
- if $s \models \neg\texttt{egg}$ then next(putIntoBowl, s) $= s$.

Then $\kappa_{\texttt{putIntoBowl}}$ (next(putIntoBowl, s)$|s$) $= 0$ and for all $s' \neq s$, $\kappa_{\texttt{putIntoBowl}}(s'|s) = +\infty$. The action theory corresponding to putIntoBowl is

$$\Sigma_{\texttt{putIntoBowl}} = \mathbf{K}\left(\neg\texttt{egg}_{t+1} \wedge \left(\bigwedge_i \texttt{in}(i+1)_{t+1} \leftrightarrow \texttt{in}(i)_t\right)\right.$$
$$\left. \wedge (\texttt{spoiled}_{t+1} \leftrightarrow (\texttt{spoiled}_t \vee \texttt{g}_t)) \wedge (\texttt{g}_{t+1} \leftrightarrow \texttt{g}_t)\right)$$

Let us now consider the BBP

$$\pi = (\texttt{takeNewEgg}; \texttt{testEgg}; \textbf{if } \mathbf{B}_1\texttt{g} \textbf{ then } \texttt{putIntoBowl})^3$$

(where $(\pi')^3$ means that the subplan π' is repeated three times) and the initial belief state Init $=$ **Only**($\mathbf{K}\texttt{in}(0)$). Figure 1 shows the progression of Init by π.

Let us give some intuitive explanations about why these 4 belief states are obtained as possible outcomes of the program:

Case 1 The three tests came out to be negative, and therefore no egg has been put into the bowl: the final belief state is $\mathbf{K}(\texttt{in}(0))$.

Case 2 Only one of the three tests came out to be positive, and therefore one egg has been put into the bowl. In the final belief state, the agent knows for sure that there is one egg on the bowl ($\mathbf{K}(\texttt{in}(1))$; moreover the agent believes to the degree two that this egg is a good one ($\mathbf{B}_2\texttt{om}(1)$): indeed, when taking an egg out of the box, the agent has a prior belief (to the degree 1) that it is good ($\mathbf{B}_1\texttt{g}$), and after testing it, a positive result reinforces this belief up to the

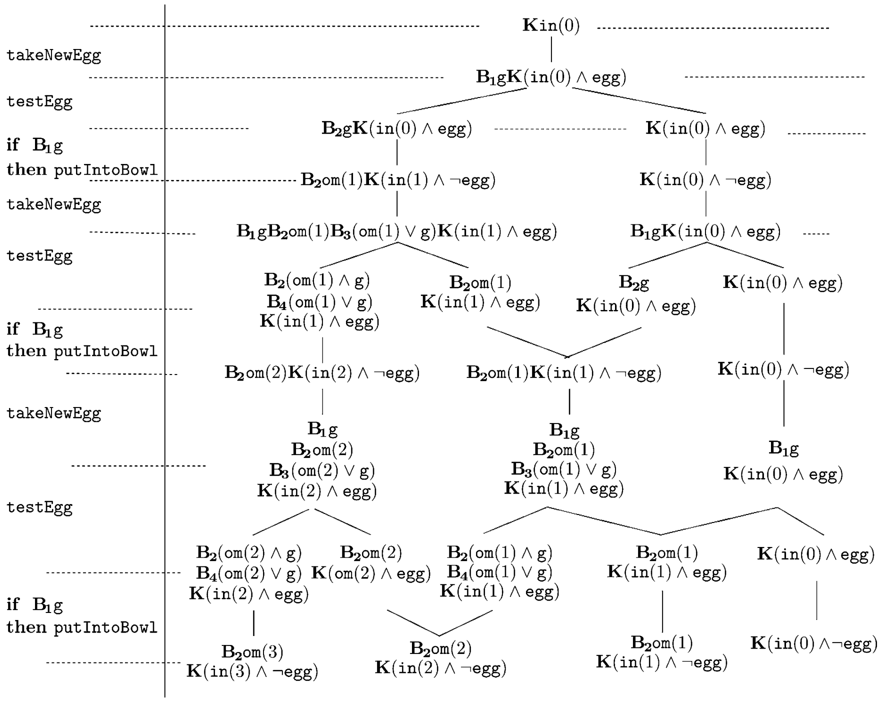

Figure 1.

degree 2, due to the reinforcement effect of combination (Section 3).

Case 3 Two out of the three tests came out to be positive: in the final belief state the agent knows that there are two eggs in the bowl, and for the same reasons as in Case 2, he believes to the degree 2 that both are good ($\mathbf{B}_2\mathtt{om}(2)$).

Case 4 All three tests being positive, the eggs have all been put into the bowl. For the same reasons as above the agent believes they are all good.

6. CONCLUSION

This paper has paved the way towards building a language for programming autonomous agents with actions, sensing (observations), and graded beliefs. Beliefs are expressed in a high-level language with graded modalities. Progression (by ontic actions and sensing) can be computed directly in this high-level language. We have shown how to compute a precompiled policy from a belief-based plan, that more or less corresponds to a policy in the POMDP meaning.

At least two issues for further research are expected in a near future.

6.1. *Integrating Belief-Based Programming and Golog*

A fairly close area is that of cognitive robotics, especially the work around *Golog* and the situation calculus (e.g., Reiter 2001a), which are concerned with logical specifications of actions and programs, including probabilistic extensions and partial observability.

First, we consider extending our belief-based plans towards a full belief-based programming language that could be an extension of *Golog* (e.g., Reiter 2001a). Golog allows for logical specifications of actions and programs, including sensing, in a high-level language which, on one hand, is far more expressive than ours, since it allows for many features (quantification over actions and states, nondeterministic choice, etc.) that are absent from our purely propositional language. Knowledge-based programming is also implementable in Golog (Reiter 2001b). On the other hand, knowledge-based programming in Golog (Reiter 2001b) does not allow for graded uncertainty.

Note that there have been several probabilistic extensions of the situation calculus and Golog: (Bacchus et al. (1999)) gives an account for the dynamics of probabilistic belief states after perceiving noisy observations and performing physical actions with noisy effectors, and (Grosskreutz and Lakemeyer (2000)) consider 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. However, both lines of work consider simple branching conditions involving objective formulas, which is not suited to knowledge-based programming. As knowledge-based programming in Golog calls for an explicit knowledge modality as in (Scherl and Levesque 1993; Reiter 2001b), graded belief-based programming needs a collection of belief modalities, together with a syntactical way of making the agent's beliefs evolve after performing an action or an observation.

Thus, enriching BBPs with the highly expressive features of Golog and the situation calculus will ultimately results in a sophisticated language for belief-based programming, which is definitely an objective we want to pursue.

6.2. *Off-line Reasoning: Introducing Second-Order Uncertainty*

An issue that has not been considered here is the off-line evaluation of belief based plans. Many problems are raised by such an issue. First, in order to represent complex sensing actions, observations should be attached with their likelihood of occurrence (as in Boutilier et al. 1998). Given this, the projection of an initial belief state by a plan needs to introduce *second-order uncertainty*: for instance, on Example 7, given that accurate observations are more frequent than inaccurate, one would like to obtain that after asking two persons, then normally the agent is in the belief state $\mathbf{O}_2 r$ or in the belief state $\mathbf{O}_2 \neg r$, and exceptionally he is in the void belief state. This calls for introducing belief states over belief states and a second family of modalities. These issues are investigated in the companion paper (Laverny and Lang 2005).

ACKNOWLEDGEMENTS

We are indebted to the two anonymous referees for their fruitful remarks, which helped us a lot rewriting this paper.

APPENDIX: OCF COMBINATION AND DEMPSTER'S RULE OF COMBINATION

A *mass assignment* is a mapping from $2^S \setminus \{\emptyset\}$ to [0, 1] such that $\sum_{X \subseteq S} m(X) = 1$. Let m_1 and m_2 be two mass assignments, then Dempster's rule combines m_1 and m_2 into the mass assignment $m_1 \oplus m_2$ defined by

$$(m_1 \oplus m_2)(X) = \frac{\sum\{m_1(Y).m_2(Z) \text{ s.t. } Y, Z \subseteq S, X = Y \cap Z\}}{1 - \sum\{m_1(Y).m_2(Z) \text{ s.t. } Y, Z \subseteq S, Y \cap Z = \emptyset\}}$$

(and undefined when there is no pair (Y, Z) such that $Y \cap Z \neq \emptyset$ and $m_1(Y).m_2(Z) \neq 0$).

Now, let ε be an infinitesimal. To every OCF κ we associate the following family $MA(\kappa)$ of infinitesimal mass assignments, defined by: for all $s \in S$ such that $\kappa(s) \neq +\infty, m(\{s\}) = o(\varepsilon^{\kappa(s)})$; for all $s \in S$ such that $\kappa(s) = +\infty, m(\{s\}) = 0$; and for any non-singleton subset X of S, $m(X) = 0$. Clearly, for any infinitesimal mass assignment m giving a zero mass to all non-singleton subsets there is exactly one

κ such that $m \in MA(\kappa)$, and we then note $OM(m) = \kappa$ (where OM stands for *order of magnitude*). Then we have the following:

PROPOSITION 9. If $OM(m_1) = \kappa_1$ and $OM(m_2) = \kappa_2$ then $OM(m_1 \oplus m_2) = \kappa_1 \oplus \kappa_2$ (and $m_1 \oplus m_2$ is undefined iff $\kappa_1 \oplus \kappa_2$ is undefined).

Proof. Let m_1 and m_2 such that $OM(m_1) = \kappa_1$ and $OM(m_2) = \kappa_2$, with $\min(\kappa_1 + \kappa_2) \neq +\infty$. Since $m_1(X) = m_2(X) = 0$ for any non-singleton X, $m_1 \oplus m_2(X) = 0$. Then

$$1 - \sum \{m_1(Y).m_2(Z) \text{ s.t. } Y, Z \subseteq S, Y \cap Z = \emptyset\}$$
$$= 1 - \sum \{m_1(\{s\}).m_2(\{s$$

$\neg r$), $\mathbf{B}_1(a_3 \to r)$ and $\mathbf{B}_1(a_4 \to \neg r)$ – therefore, a_1 means that Hans has a strong belief that r holds, etc. – and the possible observations sent as feedback by the asking action are $\mathbf{Only}(\mathbf{K}a_i)$ for $i \in \{1, \ldots, 5\}$. This ensures that (a) the answer given is always the same, and (b) no reinforcement occurs: if for instance, the agent observes a_3 many times, then $\mathbf{B}_1 r$ holds but $\mathbf{B}_2 r$ does not.

[5] In the particular case where α is deterministic, its effects can be described by *successor state axioms* of the form $x_{t+1} \leftrightarrow \varphi_t$ for each $x \in PS$.

REFERENCES

Aucher, G.: 2004, 'A combined system for update logic and belief revision', in *7th Pacific Rim Int. Workshop on Multi-Agents (PRIMA2004)*.

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

Baral, C. and J. Lobo: 1997, 'Defeasible specifications in action theories', in *Proceedings of IJCAI'97*.

Benferhat, S., D. Dubois, and H. Prade: 2001, 'A computational model for belief change and fusing ordered belief bases', in M. A. Williams and H. Rott, (eds.), *Frontiers in Belief Revision*, Kluwer Academic Publishers, pp. 109–134.

Bonet, B. and H. Geffner: 2001, 'Planning and control in artificial intelligence. A unifying perspective', *Applied Intelligence* **3**(14), 237–252.

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

Boutilier, C., R. Brafman, H. Hoos, and D. Poole: 1999, 'Reasoning with conditional *ceteris paribus statements*', in *Proceedings of the 15th Conf. on Uncertainty in Artificial Intelligence (UAI'99)*, pp. 71–80.

Boutilier, C., N. Friedman, and J. Halpern: 1998, 'Belief revision with unreliable observations', in *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pp. 127–134.

Chan, H. and A. Darwiche: 2003, 'On the revision of probabilistic beliefs using uncertain evidence', in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*.

Darwiche, A., and J. Pearl: 1997, 'On the logic of iterated belief revision', *Artificial Intelligence* **87**(1–2), 1–29.

Fariñas del Cerro, L., and A. Herzig: 1991, 'Modal logics for possibility theory', in *Proceedings of the First International Conference on the Fundamentals of AI Research (FAIR'91)*, Springer Verlag.

Dempster, A. P.: 1967, 'Upper and lower probabilities induced by a multivaluated mapping', in *Annals Mathematics Statistics* **38**, 325–339.

Dubois, D., J. Lang, and H. Prade: 1994, 'Possibilistic logic', in D. M. Gabbay, C. J. Hogger, and J. A. Robinson (eds.), *Handbook of logic in Artificial Intelligence and Logic Programming*, volume 3, Clarendon Press – Oxford, pp. 439–513.

Dubois, D., and H. Prade: 1997, 'A synthetic view of belief revision with uncertain inputs in the framework of possibility theory', *International Journal of Approximate Reasoning* **17**(2–3), 295–324.

Fagin, R., J. Halpen, Y. Moses, and M. Vardi: 1995, *Reasoning About Knowledge*, MIT Press.

Giunchiglia, E., J. Lee, N. McCain, V. Lifschitz, and H. Turner: 2003, 'Nonmonotonic causal theories', *Artificial Intelligence* **153**, 49–104.

Goldszmidt, M. and J. Pearl: 1992, 'Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions', in *Proceedings of KR'92*, pp. 661–672.

Grosskreutz, H. and G. Lakemeyer: 2000, 'Turning high-level plans into robot programs in uncertain domains', in *Proc. ECAI-2000*, pp. 548–552.

Herzig, A., J. Lang, D. Longin, and Th. Polacsek: 2000, 'A logic for planning under partial observability', in *AAAI-00*, pp. 768–773.

Herzig, A., J. Lang, and P. Marquis: 2003, 'Action representation and partially observable planning in epistemic logic', in *Proceedings of IJCAI03*, pp. 1067–1072.

Herzig, A., J. Lang, and T. Polacsek: 2001, 'A modal logic for epistemic tests', in *Proceedings of ECAI'2000*, pp. 553–557.

Kaelbling, L. P., M. L. Littman, and A. R. Cassandra: 1998, 'Planning and acting in partially observable stochastic domains', *Artificial Intelligence* **101**, 99–134.

Lang, J., P. Liberatore, and P. Marquis: 2003, 'Propositional independence : Formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, **18**, 391–443.

Lang, J., P. Marquis, and M.-A. Williams: 2001, 'Updating epistemic states', in Springer-Verlag (ed.), Lectures Notes in Artificial Intelligence 2256, *Proceedings of 14th Australian Joint Conference on Artificial Intelligence*, pp. 297–308.

Laverny, N. and J. Lang: 2004, 'From knowledge-based programs to graded BBPs, part I: on-line reasoning', in *Proceedings of ECAI-04*, pp. 368–372.

Laverny, N. and J. Lang: 2004, 'From knowledge-based programs to graded BBPs, part II: off-line reasoning', in *Proceedings of IJCAI-05*.

Levesque, H.: 1996, 'What is planning in the presence of sensing?', in *AAAI 96*, pp. 1139–1146.

Levesque, H. and G. Lakemeyer: 2000, *The Logic of Knowledge Bases*, MIT Press.

Lin, F.: 1996, 'Embracing causality in specifying the indeterminate effects of actions', in *Proc. of AAAI'96*.

Lin, F. and R. Reiter: 1994, 'Forget it!, in *Proceedings of the AAAI Fall Symposium on Relevance*, New Orleans, pp. 154–159.

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

Reiter, R.: 2001b, 'On knowledge-based programming with sensing in the situation calculus', *ACM Transactions on Computational Logic* **2**, 433–457.

Scherl, R. B. and H. J. Levesque: 1993, 'The frame problem and knowledge-producing actions', in *AAAI-93*, pp. 698–695.

Shapiro, S., M. Pagnucco, Y. Lesperance, and H. Levesque: 2000, 'Iterated belief change in the situation calculus', in *Proceedings of KR2000*, pp. 527–537.

Spohn, W.: 1988, 'Ordinal conditional functions: a dynamic theory of epistemic states', in William L. Harper and Brian Skyrms (eds.), *Causation in Decision, Belief Change and Statistics*, volume 2, Kluwer Academic Pub., pp. 105–134.

Thielscher, M.: 2001, 'Planning with noisy actions (preliminary report)', in M. Brooks, D. Powers, and M. Stumptner (eds.), *Proceedings of the Australian Joint Conference on Artificial Intelligence*, LNAI, Adelaide, Australia, December 2001, Springer.

van der Hoek, W. and J.-J.Ch. Meyer: 1991, 'Graded modalities for epistemic logic', *Logique et Analyse* **133–134**, 251–270.

van Ditmarsch, H.: 2004, *Prolegomena to Dynamic Belief Revision*. Technical report, University of Otago, New Zealand.

van Linder, B., W. van der Hoek, and John-Jules Ch. Meyer: 1994, 'Tests as epistemic updates', in *Proceedings of ECAI 1994*, pp. 331–335.

Williams, M.-A: 1994, 'Transmutations of knowledge systems', in *Proceedings of KR'94*, pp. 619–629.

Noël Laverny
IRIT, Université Paul Sabatier
31062 Toulouse Cedex
France
E-mail: Noel.Laverny@freesbee.fr

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