Updating epistemic states

Jérôme Lang[1], Pierre Marquis[2], and Mary-Anne Williams[3]

[1] IRIT - Université Paul Sabatier -Toulouse, France
[2] CRIL - Université d'Artois - Lens, France
[3] University of Newcastle - Newcastle, Australia

Abstract. Belief update is usually defined by means of operators acting on belief sets. We propose here belief update operators acting on epistemic states which convey much more information than belief sets since they express the relative plausibilities of the pieces of information believed by the agent. In the following, epistemic states are encoded as rankings on worlds. We extend a class of update operators (dependency-based updates) to epistemic states, by defining an operation playing the same role as knowledge transmutations [21] do for belief revision.

1 Introduction

While belief revision is meant to integrate new knowledge about a static world, belief update is usually thought of as taking account of a piece of information representing the effect of an evolution of the world (which may be caused by an event or an action) [13]. It has been shown in many places (e.g., [2] [5]) that iterated applications of belief revision operations need a representation of initial beliefs more informative than flat belief sets, namely, *epistemic states*. A (flat) belief set is a closed logical theory, which, when the language is propositional and generated by a finite number of propositional symbols (which is assumed here), is equivalently expressed by a propositional formula. An epistemic state is a full encoding of what the agent believes and how she is likely to revise her beliefs accordingly, which calls for a gradation of beliefs. This gradation is usually expressed by a preorder on formulas, i.e., a reflexive and transitive relation, or more specifically by a ranking function on formulas.

Oddly enough, while the distinctions between revision and update have been extensively studied, as well as postulates and strategies for iterated belief revision, the KR community has devoted much less attention on iterated belief update (exceptions being [9], [17] and [19]) and even less on update on epistemic states. This raises the following questions, in order.
1. *Is iterated belief update as worth investigating as iterated belief revision?*
2. *Can usual update operators, mapping a pair (belief set, input formula) to a belief set, be applied iteratively without trivialization?*
3. *If so, does iteration sometimes need belief update operators acting on epistemic states rather than on belief sets?*

The answer to Question 1. is obvious when one thinks of a belief update operator as a tool for computing the effects of an action given the initial beliefs of the agent. Actions are meant to be performed in sequence, especially when it comes to planning, and in this context, iterate dupdates naturally arise.

Question 2. is more complex. We propose the following answer. The process of belief revision *acting on belief sets* is not Markovian, because revising a belief set by an input formula will not lead to the same result whether it comes from a certain sequence of revision or from another one. This can be overcome either by storing the whole history of the revision process or by representing knowledge by epistemic states, which actually amounts to the same kind of information (see [14] for a general discussion), namely, not only the pure beliefs are stored but also the way they should be revised; now, belief revision *acting on epistemic states or on sequences of belief sets* can be seen as Markovian. That belief revision on belief sets should not be a Markovian process is not surprising, since belief revision is concerned with a static world and "old" beliefs still play an important role since they bear on the very same world as new ones. The latter intuition does not carry on to belief update, or at least not with the same strength. Indeed, since iterated belief updates amount to performing successive actions, the obtained belief states represent the beliefs after each action is performed, and considering this process as Markovian is generally harmless.

Now, the paper could well stop at this point, since updating epistemic states could be seen as a formal exercise with no other interest than making students work with worlds, formulas, rankings and so on. As the reader expects, this is not the case, and here is our answer to Question 3: *there are some contexts where belief update on flat belief sets is insufficient.* We give here two such contexts:

Context 1: *Successive applications of belief revisions and updates*: Planning with nondeterministic actions in *partially observable environments* calls for plans that interleave "traditional" (or *ontic*) actions acting on the world only and *knowledge-gathering* (or *epistemic*) actions that do not change the state of the world, but the beliefs of the agent, only – their role is to render the agent informed enough so as to help her choose what to do next. Therefore, it may well be the case that a belief update will be followed by a revision, or even a sequence of revisions, which obviously calls for the need of working on rich structures such as epistemic states rather than on flat belief sets.

Example 1 (*Saturday night shooting*).
Bill is a good shooter but he is sometimes drunk. When he is not drunk, shooting at a turkey results in the turkey being dead. When he is drunk, however, shooting at a turkey results in the turkey hiding. Today, Bill does not look drunk – so that in the initial belief it is more plausible, yet not totally certain, that he's not drunk; the turkey is initially alive and not hidden (and these latter beliefs are certain). Bill shoots, which may be expressed by updating the initial belief by the formula $(\mathtt{drunk} \wedge \mathtt{hidden}) \vee (\neg\mathtt{drunk} \wedge \neg\mathtt{alive})$, *with the further constraint that* $\mathtt{drunk}$ *(as well as* $\neg\mathtt{drunk}$*) cannot be changed by the action of shooting. After a*

few seconds, one hears the turkey goobling, which leads to a revision by `alive`*. Is the turkey hidden in the final state?*
With belief change operators on flat belief sets, the initial belief is `alive` $\wedge\neg$`hidden` $\wedge\neg$`drunk`, which after update according e.g. to Forbus' operator leads to the belief set $\neg$`drunk` $\wedge\neg$`hidden` $\wedge\neg$`alive`; after revision by `alive`, the new belief set is $\neg$`drunk` $\wedge\neg$`hidden` $\wedge$`alive` while the intended result is `drunk` $\wedge$`hidden` $\wedge$`alive`.

Context 2: *Evaluation of the satisfaction of a goal after a sequence of updates.* After a sequence of updates, we may want to evaluate to what point a given goal is satisfied (this is typically looked for in decision-theoretic planning). What we want at the end is an epistemic state where the worlds violating the goals are the least entrenched ones, which needs of course to work on epistemic states. Consider the same example as above, with the goal of having the turkey dead at the end, and suppose that we have also the action `bomb` always resulting in the turkey being dead – thus performing `bomb` amounts to updating by $\neg$`alive`. We should be able to conclude that the plan `shoot` normally succeeds but sometimes fails, while the plan `bomb` always succeed.

We choose to model epistemic states by ranking functions on the set of propositional worlds, or *Ordinal Conditional Functions* (OCF) – sometimes called *kappa* functions. This model is among the simplest ones, and it is frequently chosen for modelling epistemic states. For computational efficiency reasons, OCFs will not be represented explicitly but by a more compact way, namely, by means of *stratified belief bases* that induce a full OCF (see for instance [22]).

In Section 2, we give the necessary background about OCFs and stratified belief bases; next, we give the necessary background about dependency-based update. In Section 3, we show how epistemic states are updated, not only by single propositional formulas but more generally by pairs consisting of a propositional formula and a rank, and we thus propose a counterpart to belief update of what is known under the name of *transmutation* for belief revision. We proceed first by extending the notion of variable forgetting to epistemic states, and then we are in a position to define transmutations for belief update. We briefly show that our update operators on epistemic states are relevant for reasoning about action, and we conclude by discussing related work.

2 Background and notations

Let VAR be a finite set of propositional variables and $\mathcal{L}_{VAR}$ the propositional language built upon these variables and the usual connectives. For every $X \subseteq VAR$, $\mathcal{L}_X$ denotes the sublanguage of $\mathcal{L}_{VAR}$ generated from the variables of X only. For every formula φ of $\mathcal{L}_{VAR}$, $Var(\varphi)$ is the set of variables occurring in φ. $\varphi_{x\leftarrow 1}$ (resp. $\varphi_{x\leftarrow 0}$) denotes the formula from $\mathcal{L}_{VAR}$ obtained by substituting in a uniform way the variable $x \in VAR$ by the boolean constant $\top$ (resp. $\perp$) in φ. Full instantiations of variables of VAR are called worlds, and are denoted by ω, ω' etc. Full instantiations of variables of $X \subseteq VAR$ are called X-worlds, and are denoted by ω_X, ω'_X etc. 2^X denotes the set of all possible truth assignments of variables of X. $Mod(\varphi)$ is the set of models of φ, i.e., the worlds satisfying φ.

Let X and Y be two disjoint subsets of VAR, and let $\omega_X \in 2^X$, $\omega_Y \in 2^Y$. We define the concatenation of ω_X and ω_Y as the world $\omega_X \,.\, \omega_Y \in 2^{X \cup Y}$ assigning to each variable of X (resp. Y) the same value as ω_X (resp. ω_Y). If ω is a world from 2^{VAR} and $x \in VAR$ then we define $switch(\omega, x)$ as the world obtained from ω by switching the truth value of the variable x. If $X \subseteq VAR$, we say that ω and ω' agree on X, denoted by $\omega \approx_X \omega'$, if and only if ω and ω' assign the same truth value to every variable of X.

2.1 Ordinal conditional functions and stratified belief bases

Definition 1

- *An* ordinal conditional function (OCF) *r is a mapping from 2^{VAR} to $I\!N \cup \infty$. r is said to be* normalized *if and only if $\exists \omega \in 2^{VAR}$ such that $r(\omega) = 0$;*
- *A normalized OCF r induces an* entrenchment ranking *E_r on $\mathcal{L}_{VAR}$ defined by $E_r(\varphi) = \min_{\omega \models \varphi} r(\omega)$;*
- *If r and r' are two normalized OCFs, r is said to be* at least as specific *as r', noted $r \geq r'$, if and only if for every world $\omega \in 2^{VAR}$ we have $r(\varphi) \geq r'(\varphi)$.*

Unless the contrary is explicitly stated, all OCFs considered in this paper will be normalized.

The higher $r(\omega)$, the less plausible ω represents the actual state of the world. In particular, if $r(\omega) = \infty$ then ω is totally impossible. The usual interpretation of OCFs is in terms of order of magnitude of infinitesimal probabilities [20]: $r(\omega) = i < \infty$ means that the order of magnitude of the probability of ω being the actual world is in $\mathcal{O}(\varepsilon^i)$ where ε is an infinitesimal, and $r(\omega) = \infty$ if ω is an impossible world. This interpretation implies that r should be necessarily normalized.

From a practical point of view, it is not possible to ask the agent to express her beliefs under the form of a full OCF explicitly, since it is exponentially large in the number of propositional variables. Instead, it is more efficient and natural to represent them implicitly by means of stratified belief bases.

Definition 2 (stratified belief bases)

- *A* stratified belief base *(SBB) B is a finite sequence $\langle B_1, ..., B_n, B_\infty \rangle$ of propositional formulae B_i. Each i in $\{1, \ldots, n, \infty\}$ is called a* rank. *B_∞ represents fully certain beliefs, B_n the* most *entrenched among the uncertain beliefs and B_1 the* least *entrenched ones;*
- *The* cut *of level i of a SBB B is defined by $Cut(B, i) = \bigwedge_{j \geq i} B_j$; B is said to be* consistent *if and only if $Cut(B, 1)$ is consistent;*
- *The OCF r_B induced by the SBB B is defined by $\forall \omega \in 2^{VAR}, r_B(\omega) = \max\{i \mid \omega \models \neg B_i\}$ if such an index exists, and 0 otherwise; r_B is normalized if and only if B is consistent.*
- *if B is a SBB, φ a formula and i a rank in $\{1, \ldots, n, \infty\}$ then we let*

$$Add(B, \varphi, i) = \langle B_1, .., B_{i-1}, B_i \wedge \varphi, B_{i+1}, ..., B_n, B_\infty \rangle.$$

2.2 Formula-variable independence and variable forgetting

Definition 3 (FV-independence) *[16] Let φ be a formula from $\mathcal{L}_{VAR}$ and $X \subseteq VAR$. φ is said to be* independent *from X if and only if there exists a formula ψ from $\mathcal{L}_{VAR}$ logically equivalent to φ which does not mention any variable from X.*

It is shown in [16] that φ is independent from X if and only if φ is independent from $\{x\}$ for each $x \in X$; we denote by $DepVar(\varphi)$ the set of variables φ is dependent on. It is also shown in [16] that φ is independent from x if and only if $\varphi_{x\leftarrow 0}$ and $\varphi_{x\leftarrow 1}$ are logically equivalent, from which it can be derived that checking whether $x \in DepVar(\varphi)$ is coNP-complete.

The notion of variable elimination (also referred to as forgetting, projection or marginalization) is central in the following:

Definition 4 (variable forgetting) *[18] Let φ be a formula from $\mathcal{L}_{VAR}$ and $X \subseteq VAR$. $Forget(\varphi, X)$ is the formula inductively defined as follows:*
(i) $Forget(\varphi, \emptyset) = \varphi$;
(ii) $Forget(\varphi, \{x\}) = \varphi_{x\leftarrow 1} \vee \varphi_{x\leftarrow 0}$;
(iii) $Forget(\varphi, \{x\} \cup X) = Forget(Forget(\varphi, X), \{x\})$.

The following characterization of variable forgetting [15] helps to understand how it works in practice: if φ is under DNF, i.e., $\varphi = \gamma_1 \vee \ldots \vee \gamma_n$ where each γ_i is a conjunction of literals, then $Forget(\varphi, X)$ can be obtained by deleting from the γ_i's all occurrences of literals $x, \neg x$ for all $x \in X$. For instance, let $\varphi = (\neg a \vee b) \wedge (a \vee c) \wedge (b \vee c \vee d)$ and $X = \{a, d\}$. Since φ is logically equivalent to $(\neg a \wedge c) \vee (a \wedge b) \vee (b \wedge c)$, we have $Forget(\varphi, X) \equiv (b \vee c)$. $Forget(\varphi, X)$ is the strongest consequence of φ being independent from X [16].

2.3 Belief update

A *belief update operator* $\diamond$ maps the propositional belief base (a formula) K representing the initial beliefs of a given agent and an input formula α reflecting some explicit evolution of the world [13], to a new set of beliefs $K \diamond \alpha$ held by the agent after this evolution has taken place.

Katsuno and Mendelzon [13] proposed a general semantics for update. The most prominent feature of KM-updates (distinguishing updates from revision) is that update must be performed modelwise, i.e., $Mod(K \diamond \alpha) = \bigcup_{\omega \models K} \omega \diamond \alpha$. Given that updates are performed modelwise, what remains to be defined is the way models are updated, i.e., how $\omega \diamond \alpha$ is defined.

Update operators proposed in the literature can be (roughly) classified in two main families. *Minimisation-based updates* $\diamond_{Min}$ (such as Winslett's PMA [23]), stemming from the direct instantiation of the Katsuno-Mendelzon semantics, compute $\omega \diamond_{Min} \alpha$ by selecting the models of α "closest" to ω (this notion of closeness being modelled by a collection of preorders $\geq_\omega$ on 2^{VAR}). *Dependency-based updates* $\diamond_{Dep}$ ([10], [11], [6], [24], [12]) compute $\omega \diamond \alpha$ by first forgetting (from ω) the truth value of all variables that are "relevant" to α (leaving unchanged the truth value of variables not relevant to the update), and then expanding the result with α; the notion of "being relevant to" is modelled by a mapping Dep from $\mathcal{L}_{VAR}$ to 2^{VAR}. Many choices for Dep are possible (see [11] for details). The most frequent choice for Dep is *semantical dependence*: $Dep(\alpha) = DepVar(\alpha)$, and by default we let $Dep = DepVar$. Whatever the choice of Dep, the *dependence-based update* $\omega \diamond_{Dep} \alpha$ of a world ω by a formula α w.r.t. Dep is the set of all worlds ω' such that $\omega' \models \alpha$, and for every propositional variable x from VAR such that $x \notin Dep(\alpha)$, ω and ω' assign the same truth value to x.

Interestingly, $\diamond_{Dep}$ operators can be characterized through the notion of variable forgetting defined above. Indeed, the following holds [6]:

$$K \diamond_{Dep} \alpha \equiv (Forget(K, Dep(\alpha)) \wedge \alpha$$

This result gives an intuitive understanding of how dependency-based update works: first, one forgets the variables concerned by the update, and then one expands by the input.

3 Updating OCFs

We are now going to apply the principle "forget, then expand", at work in dependency-based update, to epistemic states consisting of OCFs. Therefore what we have to do first is to generalize variable forgetting to OCFs.

3.1 Independence of an OCF from a set of variables

Recall that variable forgetting can be characterized by the following result: $Forget(\varphi, X)$ is the strongest consequence of φ that is independent from X. We may thus define variable forgetting from an OCF by a similar construction, which requires first to define independence of an OCF from a set of variables.

Definition 5 *Let $X \subseteq VAR$.*

- *An OCF r is* independent *from X if and only if there is a SBB B inducing r not mentioning any variable from X.*
- *A SBB B is* independent from *X if and only if its generated OCF r_B is independent from X.*

Example 2: $B = \langle B_1, B_2, B_\infty \rangle$ *with* $B_\infty = a \rightarrow b$, $B_2 = (a \rightarrow \neg b) \wedge (a \rightarrow b \vee c) \wedge (b \rightarrow d)$, $B_1 = b$. *The OCF induced by B is the following:*

$r(\omega) = \infty$ *for each* $\omega \models a \wedge \neg b$	$r(\omega) = 2$ *for each* $\omega \models b \wedge \neg d$
$r(\omega) = 1$ *for each* $\omega \models \neg a \wedge \neg b$	$r(\omega) = 0$ *for each* $\omega \models \neg a \wedge b \wedge d$

The following simple result states that it is sufficient to focus on independence from a single variable.

Proposition 1 *r (resp. B) is independent from X if and only if r (resp. B) is independent from $\{x\}$ for all $x \in X$.*

Therefore, all the information about sets of variables an OCF r is dependent on which can be summarized by the set of variables $DepVar(r) = \{x \in VAR \mid r \text{ depends on } \{x\}\}$ (and we define $DepVar(B)$ similarly).

It is not difficult to verify that the SBB $B' = \langle B'_1, B'_2, B'_\infty \rangle$ with $B'_1 = B_1$, $B'_2 = (a \rightarrow \neg b) \wedge (b \rightarrow d)$ and $B'_\infty = B_\infty$, induces the same OCF, i.e., $r_B = r_{B'}$. Therefore, B and B' are equivalent, and since c is not mentioned in B', B is independent from $\{c\}$ (and so is the OCF r_B). On the other hand, it is dependent on $\{a\}$, $\{b\}$ and $\{d\}$, i.e., $DepVar(B) = \{a, b, d\}$.

The following result gives semantical characterizations of independence of an OCF from a variable.

Proposition 2 *Let r be an OCF and $X \subseteq VAR$. The following four statements are equivalent.*

1. *r is independent from X.*
2. *For any X-worlds $\omega_X, \omega'_X \in 2^X$, we have $r(\omega_{VAR \setminus X} \cdot \omega_X) = r(\omega_{VAR \setminus X} \cdot \omega'_X)$.*
3. *For any variable $x \in X$ and any $\omega \in 2^{VAR}$, we have $r(\omega) = r(switch(\omega, x))$.*
4. *For any nontautological φ such that $Var(\varphi) \subseteq X$, we have $E_r(\varphi) = 0$.*

In the case where the OCF is defined implicitly by a SBB, the next result gives a practical way of computing whether it is independent from $\{x\}$ without having to write r explicitly.

Proposition 3
The SBB B is independent from X if and only if for all $i \in \{1, ..., n, \infty\}$, $Cut(B, i)$ is independent from X.

Therefore, the problem of checking independence of a SBB from a variable can be reduced to a linear number of "classical" independence problems. This result enables us to draw generalizations of several results about formula-variable independence stated in [16]. In particular, determining whether B is independent from X is coNP-complete.

3.2 Forgetting in OCFs

Definition 6 *Let $X \subseteq VAR$ and r be an OCF . $Forget(r, X)$ is the minimal OCF r' (w.r.t. $\leq$) such that $r' \geq r$ and r' is independent from X.*

The following result gives a semantical characterization of forgetting.

Proposition 4 *Let r be an OCF and $X \subseteq VAR$. Then*

$$Forget(r, X)(\omega) = \min\{r(\omega') \mid \omega' \in 2^{VAR} \text{ and } \omega' \approx_{VAR \setminus X} \omega\}$$

Note that when X is a singleton $\{x\}$, the latter identity becomes $Forget(r, \{x\})(\omega) = \min(r(\omega), r(switch(\omega, x)))$. The previous definition and characterization are not operational when the OCF is represented implicitly under the form of a SBB. The next result tells us how to implement variable forgetting from a SBB in practice, namely by forgetting from the n classical propositional formulas $Cut(B, i)$.

Proposition 5 *Let B be a SBB and $X \subseteq VAR$. Let $BForget(B,X) = \langle Forget(Cut(B,i),X)\rangle_{i=1,2,\ldots,n,\infty}$. Then we have*

$$Forget(r_B, X) = r_{BForget(B,X)}$$

Example 3: Let $B = \langle B_1, B_2, B_\infty\rangle$ with $B_\infty = \top$, $B_2 = a \wedge c$, $B_1 = a \rightarrow b$. We have $BForget(B,\{a\}) = \langle b \wedge c, c, \top\rangle$; $BForget(B,\{b\}) = \langle a \wedge c, a \wedge c, \top\rangle$; $BForget(B,\{a,c\}) = \langle c, \top, \top\rangle$.

Note that it is important to take the conjunction of the strata before forgetting: since $Forget(B_2,\{a\}) = c$ and $Forget(B_1,\{a\}) = \top$, $BForget(B,\{a\})$ is not equivalent to $\langle Forget(B_1,\{a\}), Forget(B_2,\{a\}), Forget(B_\infty,\{a\})\rangle$.

3.3 Updating an epistemic state by a formula and a rank

Let's remind that a *transmutation* operator maps an OCF r, a consistent formula φ and a rank i to a new OCF $r^*(\varphi,i)$ such that $E_{r^*(\varphi,i)}(\varphi) = i$ (see [21]). On this ground, we define the update of r by α with rank i as the transmutation of $Forget(r, Dep(\alpha))$ by the new belief α together with its OCF degree i. This supposes that a transmutation operator has been previously fixed.

Definition 7 (U-transmutation)
Let $*$ be a transmutation operator, Dep a dependency function, r an OCF, α a consistent, nontautological formula and i a rank. The U-transmutation *of r by (α,i) with respect to Dep and $*$ is defined by*

$$r^\diamond(\alpha,i)(\omega) = Forget(r, Dep(\alpha))^*(\alpha,i)$$

After the forgetting process has pushed $E_{Forget(r,Dep(\alpha))}(\alpha)$ down to 0 (see last point of Proposition 2), the transmutation process pushes it up to the specified level i, i.e., enforces $E_{r^\diamond(\alpha,i)}(\alpha) = i$. Importantly, note that the higher i, the less entrenched α and the more entrenched $\neg\alpha$. Hence, when learning a new fact φ with some entrenchment degree i reflecting the evolution of the world, the initial knowledge base has to be U-transmuted by $(\neg\varphi, i)$. The higher i, the more entrenched the new information φ and the more unlikely the more plausible $\neg\varphi$-worlds. The limit case of updating by a certain input φ consists in U-transmuting by $(\neg\varphi,\infty)$ which enforces $r^\diamond(\neg\varphi,\infty)(\omega) = \infty$ for all models of $\neg\varphi$, i.e., $E_{r^\diamond(\neg\varphi,\infty)}(\neg\varphi) = \infty$.

We consider now two of the most common transmutation schemes, namely conditionalization [20] and *adjustment* [21]. The following expressions can be derived from the above definition, the general formulations of conditionalization and adjustment (omitted for the sake of brevity), and the fact that for any consistent, nontautological formula α, $E_{Forget(r,Dep(\alpha))}(\alpha) = E_{Forget(r,Dep(\alpha))}(\neg\alpha) = 0$ (last point of Proposition 2):

$* =$ conditionalization [20]

$$r^\diamond(\alpha,i)(\omega) = \begin{cases} Forget(r, Dep(\alpha))(\omega) & \text{if } \omega \models \neg\alpha \\ Forget(r, Dep(\alpha))(\omega) + i & \text{if } \omega \models \alpha \end{cases}$$

$* =$ adjustment [21]

$$r^\diamond(\alpha,i)(\omega) = \begin{cases} Forget(r, Dep(\alpha))(\omega) & \text{if } \omega \models \neg\alpha \\ \max(i, Forget(r, Dep(\alpha))(\omega)) & \text{if } \omega \models \alpha \end{cases}$$

Two limit cases are worth considering:
1. When $i = \infty$ – meaning, as said above, that the information $\neg\alpha$ is certain in the new state of affairs – then $r^\diamond(\alpha,i)(\omega)$ is independent from the choice for $*$:

$$r^\diamond(\alpha,\infty)(\omega) = \begin{cases} Forget(r, Dep(\alpha))(\omega) & \text{if } \omega \models \neg\alpha \\ \infty & \text{if } \omega \models \alpha \end{cases}$$

2. When $i = 0$, the transmutation step (whatever the choice of $*$) has no effect on $Forget(r, Dep(\alpha))$ since $E_{Forget(r,Dep(\alpha))} = 0$. This merely means that everything about the variables concerned with α has been forgotten. Note that, as a consequence, $r^\diamond(\alpha, 0)$ and $r^\diamond(\neg\alpha, 0)$ coincide and are equal to $Forget(r, Dep(\alpha))$.

Now, when the initial OCF r is given implicitly under the form of a SBB, its U-transmutation by (α, i) can be computed without generating r explicitly, in both particular cases where $*$ is a conditionalization and an adjustment.

Proposition 6 *Let B be a consistent SBB. Let r be an OCF, α a consistent, nontautological formula and i a rank.*
1. if $* =$ conditionalization then $r_B^\diamond(\alpha,i) = r_{Add(BForget(B,Dep(\alpha)),\neg\alpha,i)}$;
2. if $* =$ adjustment and $i \neq \infty$ then $r_B^\diamond(\alpha,i) = r_{ShiftAdd(BForget(B,Dep(\alpha)),\neg\alpha,i)}$ where $ShiftAdd(K,\alpha,i) = \langle K_1 \vee \neg\alpha, ..., K_{i-1} \vee \neg\alpha, K_i \wedge \alpha, (K_{i+1} \vee \neg\alpha) \wedge (K_1 \vee \alpha), ..., (K_n \vee \neg\alpha) \wedge (K_{n-i} \vee \alpha), K_{n-i+1} \vee \alpha, ..., K_n \vee \alpha, K_\infty\rangle$.

Example 4 (Door and window)
Suppose that initially, the agent knows for sure that the door is open or the window is open, and that normally *the door is open. Thus, the initial epistemic state r_B is induced by the SBB*

$B = \langle B_1 =$ `door-open`, $B_\infty =$ `door-open`$\vee$`window-open`$\rangle$

Closing the door amounts to update the epistemic state by the certain piece of information $\neg$`door-open`. *We get: (i) $DepVar(\neg$`door-open`$) = \{$`door-open`$\}$; (ii) $BForget(B,$ $\{$`door-open`$\}) = \langle\top,\top\rangle$; (iii) $r_B^\diamond($`door-open`$,\infty)$ is the OCF induced by the SBB* $\langle\top,\neg$`door-open`$\rangle$. *Now, closing the window amounts to update the epistemic state by the certain piece of information* $\neg$`window-open`, *i.e., to U-transmute it by* (`window-open`, ∞)*: (i) $DepVar(\neg$`window-open`$) = \{$`window-open`$\}$; (ii) $BForget(B, \{$`window-open`$\}) = \langle$`door-open`$,\top\rangle$;*
(iii) $r_B^\diamond($`window-open`$,\infty)$ is associated to the OCF induced by the SBB $\langle\neg$`window-open`,`door-open`$\rangle$. *Note that whereas we do not know anything more*

about the window after we closed the door, we still know that the door is normally open after we closed the window – which is intended.
Let us now consider the action "do something with the window" which results nondeterministically in the window being closed or open, none of these results being exceptional. We U-transmute the initial belief base by (`window-open`,0) *(note that, obviously, it would work as well with* ($\neg$`window-open`,0)*):* $r_B^\diamond$(`window-open`, 0) *is the OCF induced by the SBB* $\langle$`door-open`, $\top\rangle$.

3.4 Application to reasoning about action

When reasoning about action, the formula representing the knowledge about the initial state of the world is updated by the explicit changes caused by the actions. Now, it is often the case that the possible results of a nondeterministic action do not all have the same plausibility. Rather, typical nondeterministic actions have, for a given initial state, one or several normal effects, plus one or several exceptional effects, with possibly different levels of exceptionality. In this case, one has to update the initial belief base by a SBB rather than with a single formula. For the purpose of applying U-transmutations to reasoning about action, we extend U-transmutations to the case where some of the variables are not allowed to be forgotten, because they are static. We first need to partition the set of literals between *static* and *dynamic* variables, i.e., $VAR = SVAR \cup DVAR$. Static variables are persistent, i.e., their truth value does not evolve. Such a distinction is meant to forget only dynamic variables relevant to the update (static variables should not be forgotten). These static and dynamic variables may depend on the action performed and be specified together with the action description (see [12]). Note that the standard case is recovered when $SVAR = \emptyset$.

Definition 8 *The* U-transmutation *of* B *by* (α, i), *w.r.t. the static variables* $SVAR$, *a dependency relation* Dep *and a given transmutation* $*$, *is defined by* $r^\diamond(\alpha, i)(\omega) = Forget(r, Dep(\alpha) \setminus SVAR)^*(\alpha, i)$.

Example 5 *(Saturday night shooting).*
Let us consider the problem mentioned in the introduction. Let the initial epistemic state r_B *be represented by the SBB*

$$B = \langle B_1 = \texttt{alive} \wedge \neg\texttt{hidden},\ B_\infty = \neg\texttt{drunk}\rangle$$

Furthermore, `drunk` *is a static variable:* $SVAR = \{\texttt{drunk}\}$, *which means that none of the actions considered in the action model can influence the truth value of* `drunk`. *Updating by the result of the action* `shoot`, *namely*

$$\varphi = (\texttt{drunk} \rightarrow \texttt{hidden}) \wedge (\neg\texttt{drunk} \rightarrow \neg\texttt{alive})$$

gives the following result: $r_{SVAR}^\diamond(\neg\varphi, \infty)$ *is the OCF induced by the SBB*

$$\langle \neg\texttt{drunk}),\ (\texttt{drunk} \rightarrow \texttt{hidden}) \wedge (\neg\texttt{drunk} \rightarrow \neg\texttt{alive})\rangle$$

which is equivalent to (i.e., induced the same OCF) this other SBB:

$$\langle \neg\texttt{drunk} \wedge \neg\texttt{alive},\ (\texttt{drunk} \rightarrow \texttt{hidden}) \wedge (\neg\texttt{drunk} \rightarrow \neg\texttt{alive}\rangle.$$

Thus, in the final belief state, it is believed (yet with no certainty) that the turkey is dead, which is intended.

4 Related work

Changing epistemic states has been considered many times for *belief revision*, especially when it comes to iteration. In particular, the recent work of Benferhat, Konieczny, Papini and Pino-Perez [1] investigates the revision of an epistemic state by an epistemic state.

As to belief update, the closest approach to ours is Boutilier's generalized update [3]. Generalized update is more general than both belief revision and belief update. It models epistemic states by OCFs. A generalized update operation considers (i) the (explicit) description of the initial epistemic state; (ii) the dynamics of a given set of events (each of which having its own plausibility rank) expressed by a collection of transition functions mapping an initial and a final world to a rank; (iii) a formula representing an observation made after the evolution of the dynamic system; now, the output consists of the identification of the events that most likely occurred, a revised initial belief state and an updated new belief state. In the absence of observations (i.e., when updating by $\top$), generalized update merely computes the most likely evolution of the system from its dynamic and the initial belief state, which is not far from the goals of our approach. The crucial difference is in the way this most likely evolution is computed: in [3] epistemic states are represented explicitly (by fully specified ordinal conditional functions), while in our approach the dynamics of the system is represented in a very compact way: requiring that fluents dependent (resp. independent) of the input formula be forgotten (resp. remain unchanged) is a compact way to encode the dynamics of the system – it is a kind of a solution to the frame problem. In further work we plan to integrate observations (as in generalized update) in our model, and thus develop an efficient way, based on dependence relations, of performing generalized update.

Another related line of work is [7] who show that Lewis' imaging operations can be viewed as belief updates on belief states consisting of probability distributions. They propose a counterpart of imaging to possibility theory. Both classes of operations map a belief state and a flat formula to a belief state, and they are based on minimization.

Acknowledgements

The second author has been partly supported by the Région Nord/Pas-de-Calais, the IUT de Lens and the European Communities.

References

1. Salem Benferhat, Sébastien Konieczny, Odile Papini and Ramon Pino-Perez. Iterated revision by epistemic states: axioms, semantics and syntax. *Proceedings of ECAI'2000*, 13-17.
2. Craig Boutilier. Revision sequences and nested conditionals. *Proceedings of IJCAI'93*, 519-525.

3. Craig Boutilier. Generalized update: belief change in dynamic settings. *Proceedings of IJCAI'95*, 1550-1556.
4. Gerhard Brewka and Joachim Hertzberg. How to do things with worlds: on formalizing actions and plans. *Journal of Logic and Computation*, 3 (5), 517-532, 1993.
5. Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Proceedings of TARK'94*, 5-23.
6. Patrick Doherty and Witold Lukaszewicz and Ewa Madalinska-Bugaj. The PMA and relativizing change for action update. *Proceedings of KR'98*, 258-269.
7. Didier Dubois and Henri Prade. Belief revision and updates in numerical formalisms. *Proceedings of IJCAI'93*, 620-625.
8. Hélène Fargier, Jérôme Lang and Pierre Marquis. Propositional logic and one-stage decision making. *Proceedings of KR'2000*, 445-456.
9. Nir Friedman and Joseph Halpern. Modeling beliefs in dynamic systems. Part 2: revision and update, *Journal of Artificial Intelligence Research*, 10, 117-167, 1999.
10. Andreas Herzig. The PMA revisited. *Proceedings of KR'96*, 40-50.
11. Andreas Herzig and Omar Rifi. Propositional belief base update and minimal change. *Artificial Intelligence*, 115(1), 107-138, 1999.
12. Andreas Herzig, Jérôme Lang, Pierre Marquis and Thomas Polacsek. Actions, updates and planning. *Proceedings of IJCAI'2001*, 119-124.
13. Hirofumi Katsuno and Alberto Mendelzon. On the difference between updating a knowledge base and resvising it. *Artificial Intelligence* 52:263-294, 1991.
14. Sébastien Konieczny. *Sur la logique du changement : révision et fusion de bases de connaissances.* Thèse de l'Université des Sciences et Technologies de Lille, 1999. In French.
15. Jérôme Lang, Paolo Liberatore and Pierre Marquis. Complexity results for propositional independence. In preparation.
16. Jérôme Lang and Pierre Marquis. Complexity results for independence and definability in propositional logic. *Proceedings of KR'98*, 356-367.
17. Shai Berger, Daniel Lehmann and Karl Schlechta. Preferred history semantics for iterated updates. *Journal of Logic and Computation*, 9(6), 817-833.
18. Fangzhen Lin and Ray Reiter. Forget it! *Proceedings of the AAAI Fall Symposium on Relevance*, New Orleans, 1994, 154-159.
19. Steven Shapiro, Maurice Pagnucco, Yves Lespérance and Hector Levesque. Iterated change in the situation calculus. *Proceedings of KR'2000*, 527-538.
20. W. Spohn. Ordinal conditional functions: a dynamic theory of epistemic states. In *Causation in Decision, Belief Change, and Statistics*, vol. 2, 105-134, Kluwer Academic Publishers, 1987.
21. Mary-Anne Williams. Transmutations of knowledge systems. *Proceedings of KR'94*, 619-629.
22. Mary-Anne Williams. Iterated theory base change: a computational model. *Proceedings of IJCAI'95*, 1541-1547.
23. Marianne Winslett. Reasoning about action using a possible model approach. *Proceedings of AAAI'88*, 89-93.
24. Yan Zhang and Normal Foo. Reasoning about persistence: a theory of actions. *Proceedings of IJCAI'93*, 718-723.