An ASM Refinement and Implementation of the CONDOR System using Ordinal Conditional Functions

Christoph Beierle[1], Gabriele Kern-Isberner[2]

[1]Dept. of Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany
[2]Dept. of Computer Science, University of Dortmund, 44221 Dortmund, Germany
christoph.beierle@fernuni-hagen.de
gabriele.kern-isberner@cs.uni-dortmund.de

Abstract. A high-level ASM specification of CONDOR, a system providing powerful methods for managing beliefs induced by general conditionals, is refined to qualitative conditional logics, leading to an ASM implementation. The semantics of this logic are given by ordinal conditional functions (OCFs) ordering worlds according to their plausibility. OCFs are used to represent the epistemic state of an agent carrying out knowledge management tasks like answering queries, performing diagnosis and hypothetical reasoning, or revising and updating its own state of belief in the light of new information. The ASM methodology allows us to precisely describe these knowledge management operations and clarify various subtleties.

1 Introduction

The aim of the CONDOR system is to provide powerful methods and tools for managing beliefs induced by general conditionals. Figure 1 provides a bird's-eye view of the CONDOR system. CONDOR can be seen as an agent being able to take rules, pieces of evidence, queries, etc., from the environment and giving back sentences it believes to be true with a degree of certainty. Basically, these degrees of belief are calculated from the agent's current epistemic state which is a representation of its cognitive state at the given time. The agent is supposed to live in a dynamic environment, so it has to adapt its epistemic state constantly to changes in the surrounding world and to react adequately to new demands (cf. [1], [9]).

In [4] we developed a high-level ASM specification CONDORASM for the system. Thereby, we were able to elaborate crucial interdependencies between different aspects of knowledge representation, knowledge discovery, and belief revision. However, in [4], we deliberately left various universes and functions of CONDORASM abstract, aiming at a broad applicability of our approach. In

The research reported here was partially supported by the DFG – Deutsche Forschungsgemeinschaft (grant BE 1700/5-3).

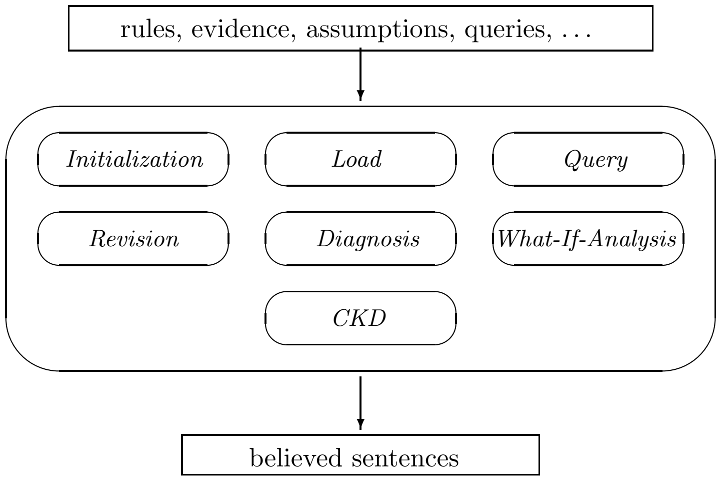

Fig. 1. A bird's-eye view of the Condor-Systems

particular, by leaving the universe $\mathcal{Q}$ of quantitative and qualitative scales abstract, the specification applies to conditionals in both quantitative logic (such as probabilistic logic [24, 25]) and qualitative approaches (like ordinal conditional functions [27]).

In this paper, we will refine CondorASM to qualitative conditionals whose semantics is given by ordinal conditional functions (OCFs) ordering possible worlds according to their plausibility. In the resulting CondorASM$_{\mathcal{O}}$, OCFs are used to represent the epistemic state of an agent carrying out knowledge management tasks like answering queries, performing diagnosis and hypothetical reasoning, or revising and updating its own state of belief in the light of new information. The ASM methodology allows us to precisely describe these knowledge management operations and clarify various subtleties. Moreover, by carrying out all required refinement steps, we arrived at an operational model [23] that has been impemented in XASM [2]. For instance, all example computations given in this paper have been carried out by the CondorASM implementation given in [23].

The rest of this paper is organized as follows: In Section 2, we recall the basics of qualitative conditional logics and fix our notation, and in Section 3, the formal framework of CondorASM$_{\mathcal{O}}$ is introduced. Section 4 summarizes the overall structure of CondorASM$_{\mathcal{O}}$ and decribes the realization of all its top-level functions in the OCF framework. Section 5 contains some conclusive remarks and points out further work.

2 Qualitative Conditional Logic

We start with a propositional language $\mathcal{L}$, generated by a finite set Σ of atoms $a, b, c, \ldots$. The formulas of $\mathcal{L}$ will be denoted by uppercase roman letters

$A, B, C, \ldots$. For conciseness of notation, we will omit the logical *and*-connector, writing AB instead of $A \wedge B$, and barring formulas will indicate negation, i.e. $\overline{A}$ means $\neg A$. Let Ω denote the set of possible worlds over $\mathcal{L}$; Ω will be taken here simply as the set of all propositional interpretations over $\mathcal{L}$ and can be identified with the set of all complete conjunctions over Σ. $\omega \models A$ means that the propositional formula $A \in \mathcal{L}$ holds in the possible world $\omega \in \Omega$.

By introducing a new binary operator $|$, we obtain the set $(\mathcal{L} \mid \mathcal{L}) = \{(B|A) \mid A, B \in \mathcal{L}\}$ of conditionals over $\mathcal{L}$. $(B|A)$ formalizes "*if A then B*" and establishes a plausible, probable, possible etc connection between the *antecedent* A and the *consequent* B. Here, conditionals are supposed not to be nested, that is, antecedent and consequent of a conditional will be propositional formulas.

A conditional $(B|A)$ is an object of a three-valued nature, partitioning the set of worlds Ω in three parts: those worlds satisfying AB, thus *verifying* the conditional, those worlds satisfying $A\overline{B}$, thus *falsifying* the conditional, and those worlds not fulfilling the premise A and so which the conditional may not be applied to at all. This allows us to represent $(B|A)$ as a *generalized indicator function* going back to [10] (where u stand for *unknown* or *indeterminate*):

$$(B|A)(\omega) = \begin{cases} 1 & \text{if} \quad \omega \models AB \\ 0 & \text{if} \quad \omega \models A\overline{B} \\ u & \text{if} \quad \omega \models \overline{A} \end{cases}$$

To give appropriate semantics to conditionals, they are usually considered within richer structures such as *epistemic states*. Besides certain (logical) knowledge, epistemic states also allow the representation of preferences, beliefs, assumptions etc of an intelligent agent. Basically, an epistemic state allows one to compare formulas or worlds with respect to plausibility, possibility, necessity, probability etc.

Well-known qualitative, ordinal approaches to represent epistemic states are Spohn's *ordinal conditional functions, OCFs*, (also called *ranking functions*) [27], and *possibility distributions* [6], assigning degrees of plausibility, or of possibility, respectively, to formulas and possible worlds. In such qualitative frameworks, a conditional $(B|A)$ is valid (or *accepted*), if its confirmation, AB, is more plausible, possible etc. than its refutation, $A\overline{B}$; a suitable degree of acceptance is calculated from the degrees associated with AB and $A\overline{B}$.

3 The formal framework of CondorASM$_{\mathcal{O}}$

After introducing the universes of CondorASM$_{\mathcal{O}}$, we point out crucial aspects of revising epistemic states and describe the main properties of the central notion of *c-revision* as developed in [22].

3.1 Universes

The universe Σ of propositional variables provides a vocabulary for denoting simple facts. The universe Ω contains all possible worlds that can be distinguished using Σ. $Fact_{\mathcal{U}}$ is the set of all (unquantified) propositional sentences over Σ, i.e. $\mathit{Fact}_{\mathcal{U}}$ consists of all formulas from $\mathcal{L}$. The set of all (unquantified) conditional sentences from $(\mathcal{L} \mid \mathcal{L})$ is denoted by $\mathit{Rule}_{\mathcal{U}}$.

The universe of all sentences without any qualitative or quantitative measure is given by

$$\mathit{Sen}_{\mathcal{U}} = \mathit{Fact}_{\mathcal{U}} \cup \mathit{Rule}_{\mathcal{U}}$$

with elements written as A and $(B|A)$, respectively. Additionally, $\mathit{SimpleFact}_{\mathcal{U}}$ denotes the set of simple facts $\Sigma \subseteq \mathit{Fact}_{\mathcal{U}}$, i.e. $\mathit{SimpleFact}_{\mathcal{U}} = \Sigma$.

In CONDORASM$_{\mathcal{O}}$, the abstract universes of quantified sentences $\mathit{Sen}_{\mathcal{Q}} = \mathit{Fact}_{\mathcal{Q}} \cup \mathit{Rule}_{\mathcal{Q}}$ of [4] are refined to

$$\mathit{Sen}_{\mathcal{O}} = \mathit{Fact}_{\mathcal{O}} \cup \mathit{Rule}_{\mathcal{O}}$$

whose elements are written as $A[m]$ and $(B|A)[m]$, respectively, where $A, B \in \mathit{Fact}_{\mathcal{U}}$ and $m \in \mathbb{N}$. For instance, the measured conditional $(B|A)[m]$ has the reading *if A then B with degree of belief m*. The set of measured simple facts is denoted by $\mathit{SimpleFact}_{\mathcal{O}} \subseteq \mathit{Fact}_{\mathcal{O}}$.

Spohn [27] uses ordinal conditional functions, OCFs, (also called ranking functions)

$$\kappa : \Omega \to \mathbb{N}$$

to express degrees of plausibility of propositional formulas by specifying degrees of disbeliefs of their negation. At least one world must be regarded as being normal; therefore, $\kappa(\omega) = 0$ for at least one $\omega \in \Omega$. Each such ranking function can be taken as the representation of a full epistemic state of an agent. Thus, the abstract universe *EpState* of [4] is refined to

$$\mathit{EpState}_{\mathcal{O}} = \{\kappa \mid \kappa : \Omega \to \mathbb{N} \text{ and } \kappa^{-1}(0) \neq \emptyset\}$$

Each $\kappa \in \mathit{EpState}_{\mathcal{O}}$ uniquely determines a function (also denoted by κ)

$$\kappa : \mathit{Sen}_{\mathcal{U}} \to \mathbb{N} \cup \{\infty\}$$

defined by

$$\kappa(A) = \begin{cases} \mathit{min}\{\kappa(\omega) \mid \omega \models A\} & \text{if } A \text{ is satisfiable} \\ \infty & \text{otherwise} \end{cases}$$

for sentences $A \in \mathit{Fact}_{\mathcal{U}}$ and by

$$\kappa((B|A)) = \begin{cases} \kappa(A \wedge B) - \kappa(A) & \text{if } \kappa(A) \neq \infty \\ \infty & \text{otherwise} \end{cases}$$

for conditionals $(B|A) \in \mathit{Rule}_{\mathcal{U}}$.

Later on in Sec. 4.4, when we introduce how to query the beliefs of an agent being in an epistemic state represented by κ, we will also define the binary satisfaction relation (modelled by its characteristic function in the ASM framework)

$$\models_{\mathcal{O}} \subseteq \mathit{EpState}_{\mathcal{O}} \times \mathit{Sen}_{\mathcal{O}}$$

such that $\kappa \models_{\mathcal{O}} (B|A)[m]$ means that the state κ satisfies the sentence $(B|A)[m]$, expressing the agent's degree of belief m in $(B|A)$.

3.2 Revision of Conditional Knowledge

Belief revision, the theory of dynamics of knowledge, has been mainly concerned with propositional beliefs for a long time. The most basic approach here is the *AGM-theory* presented in the seminal paper [1] as a set of postulates outlining appropriate revision mechanisms in a propositional logical environment. This framework has been widened by Darwiche and Pearl [9] for (qualitative) epistemic states and conditional beliefs. An even more general approach, unifying revision methods for quantitative and qualitative representations of epistemic states, is described in [20]. The crucial meaning of conditionals as *revision policies* for belief revision processes is made clear by the so-called *Ramsey test* [26], according to which a conditional $(B|A)$ is accepted in an epistemic state Ψ, iff revising Ψ by A yields belief in B:

$$\Psi \models (B|A) \quad \text{iff} \quad \Psi * A \models B \tag{1}$$

where $*$ is a belief revision operator (see e.g. [26, 8]).

Note, that the term "belief revision" is a bit ambiguous: On the one hand, it is used to denote quite generally *any* process of changing beliefs due to incoming new information [13]. On a more sophisticated level, however, one distinguishes between different kinds of belief change. For instance, *(genuine) revision* takes place when new information about a static world arrives, whereas *updating* tries to incorporate new information about a (possibly) evolving, changing world [18].

In the following, we will present a precise ASM specification of both update and (genuine) revision in the context of ordinal conditional functions. We will also show how these operations can be used to realize *focusing* [11], i.e. applying generic knowledge to the evidence present by choosing an appropriate context or reference class.

3.3 C-revisions

One of the main objectives in belief revision is, informally speaking, to adopt a state of belief to some new information while respecting the previous knowledge as faithfully as possible. Whereas for the quantitative approach of probabilistic logic the principle of minimum cross entropy provides an information theoretic optimal guideline [24], the principle of *conditional preservation* as developed in [20, 22] is a general framework applicable in both quantitative and qualitative settings. By introducing the notion of *conditional indifference*, OCF functions which represent a set $\mathcal{R}$ of conditionals and which are indifferent with respect to them are called *c-representations* [22]. Furthermore, [22] introduces the concept of *c-revision* for characterizing revisions satisfying the principle of conditional preservation.

A c-revision transforms an epistemic state and a set of quantified sentences into a new epistemic state. In order to avoid lengthy case distinctions, we assume that all sentences are conditionals which can easily be achieved by representing A by $(A|\mathit{true})$.

The basic idea of a c-revision is to faithfully respect the conditional structure (cf. [22]). A characterization theorem of [22] shows that every c-revision $\kappa * \mathcal{R}$

of an epistemic state κ and a set of rules $\mathcal{R}$ can be obtained by adding to each $\kappa(\omega)$ values for each rule $R_i \in \mathcal{R}$, depending on whether ω verfies or falsifies R_i. We will now describe a procedure from [22] how to calculate such a c-revision for any finite OCF κ and any finite consistent set $\mathcal{R}$ of conditionals.

The consistency of a set $\mathcal{R} = \{(B_1|A_1), \ldots, (B_n|A_n)\}$ of conditionals in a qualitative framework can be characterized by the notion of *tolerance*. A conditional $(B|A)$ is said to be tolerated by a set of conditionals $\mathcal{R}$ iff there is a world ω such that ω verifies $(B|A)$ (i.e. $(B|A)(\omega) = 1$) and ω does not falsify any of the conditionals in $\mathcal{R}$ (i.e. $r(\omega) \neq 0$ for all $r \in \mathcal{R}$). $\mathcal{R}$ is consistent iff there is an ordered partition $\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_k$ of $\mathcal{R}$ such that each conditional in $\mathcal{R}_m$ is tolerated by $\bigcup_{j=m}^{k} \mathcal{R}_j$, $0 \leqslant m \leqslant k$ (cf. [15]). The boolean function

$$\mathit{consistencyCheck}: \mathcal{P}(\mathit{Sen}_{\mathcal{O}}) \rightarrow \mathit{Bool}$$

tests the consistency of a set of conditionals.

Example 1. Suppose we have the propositional atoms f - *flying*, b - *birds*, p - *penguins*, w - *winged* animals, k - *kiwis*. Let the set $\mathcal{R}$ consist of the following conditionals:

$$\begin{array}{lrll} \mathcal{R} & r_1: & (f|b) & \textit{birds fly} \\ & r_2: & (b|p) & \textit{penguins are birds} \\ & r_3: & (\overline{f}|p) & \textit{penguins do not fly} \\ & r_4: & (w|b) & \textit{birds have wings} \\ & r_5: & (b|k) & \textit{kiwis are birds} \end{array}$$

The conditionals r_1, r_4, and r_5 are tolerated by $\mathcal{R}$, whereas r_2 and r_3 are not; but both r_2 and r_3 are tolerated by the set $\{r_2, r_3\}$. This yields the partitioning $\mathcal{R}_0 = \{r_1, r_4, r_5\}$, $\mathcal{R}_1 = \{r_2, r_3\}$ showing the consistency of $\mathcal{R}$.

Now suppose that $\mathcal{R}$ is consistent and that a corresponding partition $\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_k$ of $\mathcal{R}$ is given. Then the following yields a c-revision: Set successively, for each partitioning set $\mathcal{R}_m$, $0 \leqslant m \leqslant k$, starting with $\mathcal{R}_0$, and for each conditional $r_i = (B_i|A_i) \in \mathcal{R}_m$

$$\kappa_i^- \;:=\; 1 \;+\; \max_{\substack{\omega \,\models\, A_i B_i \\ r(\omega) \neq 0, \forall r \in \mathcal{R}_m \cup \ldots \cup \mathcal{R}_k}} (\kappa(\omega) \;+\; \sum_{\substack{r_j \in \mathcal{R}_0 \cup \ldots \cup \mathcal{R}_{m-1} \\ r_j(\omega) = 0}} \kappa_j^-)$$

Finally, choose κ_0 appropriately to make

$$\kappa^*(\omega) \;=\; \kappa_0 + \kappa(\omega) + \sum_{\substack{1 \leqslant i \leqslant n \\ \omega \models A_i \overline{B}_i}} \kappa_i^-$$

an ordinal conditional function. Therefore, we get a function

$$\mathit{cRevision}: \mathit{EpState}_{\mathcal{O}} \times \mathcal{P}(\mathit{Sen}_{\mathcal{Q}}) \rightarrow \mathit{EpState}_{\mathcal{O}}$$

yielding a c-revison $\mathit{cRevision}(\kappa, \mathcal{R})$ for any OCF κ and any consistent set of conditionals $\mathcal{R}$; more details can be found in [22], and in [23] a complete ASM implementation is given. In the following, we will show how c-revisions can be used for the realisation of various knowledge management functions in the CONDOR system.

4 CondorASM$_{\mathcal{O}}$

We will first summarize the overall structure of CondorASM$_{\mathcal{O}}$ and then present the realization of its functionality as indicated in Figure 1.

4.1 Overall structure

As in CondorASM of [4], also in CondorASM$_{\mathcal{O}}$ the agent's current epistemic state is denoted by the controlled nullary function[1] $\mathit{currstate} : \mathit{EpState}_{\mathcal{O}}$, and the agents beliefs returned to the environment can be observed via the controlled function $\mathit{believed\_sentences} : \mathcal{P}(\mathit{Sen}_{\mathcal{O}})$ with $\mathcal{P}(S)$ denoting the power set of S.

As indicated in Figure 1, there are seven top-level functions that can be invoked, ranging from initialization of the system to the automatic discovery of conditional knowledge (CKD). Thus, we have a universe

$$\begin{aligned}\mathit{WhatToDo} \;=\; \{\, & \mathit{Initialization}, \mathit{Load}, \mathit{Query}, \mathit{Revision},\\ & \mathit{Diagnosis}, \mathit{What\text{-}If\text{-}Analysis}, \mathit{CKD}\,\}\end{aligned}$$

The nullary interaction function $\mathit{do} : \mathit{WhatToDo}$ is set by the environment in order to invoke a particular function. We tacitly assume that *do* is reset to *undef* by CondorASM$_{\mathcal{O}}$ after each corresponding rule execution.

The appropriate inputs to the top-level functions are modelled by the following monitored nullary functions set by the environment:

input type		monitored nullary fct.
$\mathcal{P}(\mathit{Sen}_{\mathcal{O}})$	:	*rule_base*
		new_information
		assumptions
$\mathcal{P}(\mathit{Sen}_{\mathcal{U}})$	:	*queries*
		goals

input type		monitored nullary fct.
$\mathcal{P}(\mathit{Fact}_{\mathcal{O}})$	:	*evidence*
$\mathcal{P}(\mathit{SimpleFact}_{\mathcal{U}})$	:	*diagnoses*
$\mathit{EpState}_{\mathcal{O}}$	:	*stored_state*
RevisionOp	:	*rev_op*

For instance, simply querying the system takes a set of (unquantified) sentences from $\mathit{Sen}_{\mathcal{U}}$, asking for the degrees of belief for them. Similarly, the *What-If-Analysis* realizes hypothetical reasoning, taking a set of (quantified) sentences from $\mathit{Sen}_{\mathcal{O}}$ as assumptions, together with a set of (unquantified) sentences from $\mathit{Sen}_{\mathcal{U}}$ as goals, asking for the degrees of belief for these goals under the given assumptions. The specific usage of all monitored functions will be explained in detail in the following section along with the corresponding top-level functionalities.

4.2 Initialization

When creating a new agent, at first no knowledge at all might be available. We model this situation by the nullary function $\mathit{uniform} : \mathit{EpState}$ taken to initialize

[1] For a general introdcution to ASMs and also to stepwise refinement using ASMs see e.g. [16] and [28]; in particular, we will use the classification of ASM functions – e.g. into controlled or monitored functions – as given in [28].

ω	$\kappa(\omega)$	ω	$\kappa(\omega)$	ω	$\kappa(\omega)$	ω	$\kappa(\omega)$
$pbfwk$	2	$p\overline{b}fwk$	5	$\overline{p}bfwk$	0	$\overline{p}\overline{b}fwk$	1
$pbfw\overline{k}$	2	$p\overline{b}fw\overline{k}$	4	$\overline{p}bfw\overline{k}$	0	$\overline{p}\overline{b}fw\overline{k}$	0
$pbf\overline{w}k$	3	$p\overline{b}f\overline{w}k$	5	$\overline{p}bf\overline{w}k$	1	$\overline{p}\overline{b}f\overline{w}k$	1
$pbf\overline{w}\overline{k}$	3	$p\overline{b}f\overline{w}\overline{k}$	4	$\overline{p}bf\overline{w}\overline{k}$	1	$\overline{p}\overline{b}f\overline{w}\overline{k}$	0
$pb\overline{f}wk$	1	$p\overline{b}\overline{f}wk$	3	$\overline{p}b\overline{f}wk$	1	$\overline{p}\overline{b}\overline{f}wk$	1
$pb\overline{f}w\overline{k}$	1	$p\overline{b}\overline{f}w\overline{k}$	2	$\overline{p}b\overline{f}w\overline{k}$	1	$\overline{p}\overline{b}\overline{f}w\overline{k}$	0
$pb\overline{f}\overline{w}k$	2	$p\overline{b}\overline{f}\overline{w}k$	3	$\overline{p}b\overline{f}\overline{w}k$	2	$\overline{p}\overline{b}\overline{f}\overline{w}k$	1
$pb\overline{f}\overline{w}\overline{k}$	2	$p\overline{b}\overline{f}\overline{w}\overline{k}$	2	$\overline{p}b\overline{f}\overline{w}\overline{k}$	2	$\overline{p}\overline{b}\overline{f}\overline{w}\overline{k}$	0

Fig. 2. Epistemic state κ after initialization with rule set $\mathcal{R}$ from Example 1

the system. In our OCF setting, this ranking function representing complete ignorance regards all possible worlds as equally pausible, i.e. $\mathit{uniform}(\omega) = 0$ for all $\omega \in \Omega$.

If, however, default knowledge is at hand to describe the problem area under consideration, an epistemic state has to be found to appropriately represent this prior knowledge. To this end, in [4] we assumed an inductive representation method to establish the desired connection between sets of sentences and epistemic states. Whereas generally, a set of sentences S allows a (possibly large) set of models (or epistemic states), in an inductive formalism we have a function *inductive* : $\mathcal{P}(\mathit{Sen}_{\mathcal{Q}}) \rightarrow \mathit{EpState}$ that selects a unique, "best" epistemic state from all those states satisfying S.

Thus, we can initialize the system with an epistemic state by providing a set of (quantified) sentences S and generating a full epistemic state from it by inductively completing the knowledge given by S. Already here, we can use the powerful concept of c-revisions for realizing the *inductive* function: *inductive*(S) is obtained by a c-revision of the epistemic state *uniform* by S:

if $\mathit{do} = \mathit{Initialization}$
  then if $\mathit{consistencyCheck}(\mathit{rule\_base}) = \mathit{false}$
    then *output*("rule base for initialization is inconsistent")
    else $\mathit{newRulesSinceUpdate} := \mathit{rule\_base}$
      $\mathit{currstate} := \mathit{cRevision}(\mathit{uniform}, \mathit{rule\_base})$

where the the monitored nullary function $\mathit{rule\_base} : \mathcal{P}(\mathit{Sen}_{\mathcal{O}})$ is used for reading the set S. The role of the nullary function $\mathit{newRulesSinceUpdate} : \mathcal{P}(\mathit{Sen}_{\mathcal{O}})$ is to keep track of all new rules that have been used to revise the agent's epistemic state since its last update occurred. Its purpose will become clear, when we discuss the subtleties of update and genuine revision operations (Sec. 4.5).

Example 2. Given the rule set $\mathcal{R}$ from Example 1 for initialization, CondorASM$_{\mathcal{O}}$ computes the epistemic state κ as depicted in Figure 2.

Selecting a "best" epistemic state from all all those states satisfying a set of sentences S is an instance of a general problem which we call the *representation*

problem (cf. [5]). There are several well-known methods to model such an inductive formalism, a prominent one being the *maximum entropy* approach. The rationale behind this approach is to represent the knowledge given by $\mathcal{S}$ most faithfully, i.e. without adding information unnecessarily (cf. [24, 25, 19]).
In an ordinal framework, system-Z, system-Z^+ [15], or system-Z^* [14], as well as the lcd-functions presented in [7] can be used to build up rankings from default knowledge; more general approaches to solve the representation problem are described in [29] and [21].

4.3 Loading an epistemic state

As in CondorASM, also in CondorASM$_\mathcal{O}$ we can initialize the system with an epistemic state by loading such a state directly from the environment where it had been stored during a previous run of the system

$$\begin{array}{l} \textbf{if } \ do = Load \\ \quad \textbf{then } \ currstate := stored\_state \end{array}$$

with a corresponding monitored nullary function $stored\_state : EpState_\mathcal{O}$.

4.4 Querying an epistemic state

In order to define the beliefs of an agent, we first refine the abstract satisfaction relation $\models_\mathcal{Q}$ of [4] to

$$\models_\mathcal{O} \ \subseteq EpState_\mathcal{O} \times Sen_\mathcal{O}$$

where $\kappa \models_\mathcal{O} (B|A)[m]$ expresses whether the quantified sentence $(B|A)[m]$ is believed in epistemic state κ. The idea is that the degree of disbelief of AB (verifying the unquantified conditional) should be more than m smaller than the degree of disbelief of $A\overline{B}$ (falsifying the unquantified conditional). Thus, we have

$$\kappa \models_\mathcal{O} (B|A)[m] \quad \text{iff} \quad \kappa(AB) + m \ < \ \kappa(A\overline{B})$$

Note that for a propositional formula, $\kappa \models_\mathcal{O} A[m]$ is obtained from the general case of a conditional by viewing A as the conditional $(A|true)[m]$ with trivial precondition *true*. Thus, $\kappa \models_\mathcal{O} A[m]$ iff $\kappa \models_\mathcal{O} (A|true)[m]$ iff $\kappa(A) + m \ < \ \kappa(\overline{A})$.

Obviously, if $m > 0$, then $\kappa \models_\mathcal{O} (B|A)[m]$ implies $\kappa \models_\mathcal{O} (B|A)[m-1]$. We are therefore mainly interested in the maximal m such that $\kappa \models_\mathcal{O} (B|A)[m]$. Therefore, the abstract function *belief* of [4] is refined to the function

$$belief_\mathcal{O} : \ EpState_\mathcal{O} \times \mathcal{P}(Sen_\mathcal{U}) \rightarrow \mathcal{P}(Sen_\mathcal{O})$$

subject to the condition

$$belief_\mathcal{O}(\kappa, \mathcal{S}) \ = \ \{S[m] \mid S \in \mathcal{S} \text{ and } \kappa \models_\mathcal{O} S[m] \text{ and } \kappa \not\models_\mathcal{O} S[m+1]\}$$

for every $\kappa \in EpState_\mathcal{O}$ and $S \subseteq Sen_\mathcal{U}$.

For a given state κ, the call $belief_\mathcal{O}(\kappa, S)$ returns, in the form of measured sentences, the beliefs that hold with regard to the set of basic sentences $S \subseteq Sen_\mathcal{U}$. The monitored function $queries : \mathcal{P}(Sen_\mathcal{U})$ holds the set of sentences and is used in the rule:

if *do* = *Query*
  then *believed_sentences* := $belief_{\mathcal{O}}$(*currstate*, *queries*)

For simplicity reasons, we assume that propositional sentences are represented by conditionals with trivial precondition *true* so that we any have to deal with conditionals. Then $belief_{\mathcal{O}}$ is defined by:

$$
\begin{array}{l}
\mathit{belief}_{\mathcal{O}}(\kappa, \mathrm{S}) \equiv \mathit{resultSet} \; := \emptyset; \\
\qquad \textbf{do forall } (B|A) \in \mathcal{S} \\
\qquad\quad \textbf{let } r_p = \mathit{apply}(\kappa, A \wedge B) \\
\qquad\qquad r_n = \mathit{apply}(\kappa, A \wedge \overline{B}) \\
\qquad\qquad d \;= r_n - r_p \textbf{ in} \\
\qquad\qquad \textbf{if } d > 0 \\
\qquad\qquad \textbf{then } \mathit{resultSet} \; := \mathit{resultSet} \cup \{(B|A)[d-1]\}
\end{array}
$$

where $apply(\kappa, A)$ returns the rank $\kappa(A)$ of an unquantified propositional sentence A under the ordinal conditional function κ (cf. Section 3.1).

In the CondorASM$_{\mathcal{O}}$ implementation developed in [23], in addition to returning the believed sentences as specified by $belief_{\mathcal{O}}$, those sentences $(B|A)$ in S that are not believed in κ (i.e. there is no such $m \in \mathbb{N}$ with $\kappa \models_{\mathcal{O}} (B|A)[m]$) are printed out as an additional information for the user.

Example 3. When asked the query $(f|p)$ (*"Do penguins fly?"*) in the epistemic state κ obtained in Example 2, CondorASM$_{\mathcal{O}}$ tells us that $(f|p)$ does not belong to the set of believed sentences; the knowledge base used for building up κ explicitly contains the opposite rule $(\overline{f}|p)$.

On the other hand, asking $(w|k)$ (*"Do kiwis have wings?"*) we get a positive degree of belief 1: From their superclass *birds*, kiwis inherit the property of having wings.

4.5 Revising and updating an epistemic state

In belief revision, one usually distinguishes between different revision operators, such as e.g. updating an epistemic state, expanding it, or setting the focus in it to a given set of sentences. Therefore, in the general framework of [4], we used a universe *RevisionOp* of revision operators acknowledging the richness of different revision methods. For the general task of revising knowledge we used the abstract function

$$\mathit{revise}: \mathit{EpState} \times \mathit{RevisionOp} \times \mathcal{P}(\mathit{Sen}_{\mathcal{Q}}) \rightarrow \mathit{EpState}$$

where a call $revise(\Psi, op, \mathcal{S})$ yields a new state where Ψ is modified according to the revision operator *op* and the set of sentences $\mathcal{S}$.

In this section, we will refine this abstract function *revise* to the OCF setting for the two most important knowledge management functions, namely update and (genuine) revision, indicated by the revision operators *Update*, *Revision* $\in$ *RevisionOp*. The operator *Focusing* will be defined for the realization of

CondorASM$_{\mathcal{O}}$'s top-level function *Diagnosis* where in addition to its epistemic state an agent has to take into account evidential facts about a particular case.

We will now focus on the sometimes subtle differences between genuine revision and updating; for a extensive discussion we refer to [22]. Essentially, genuine revision means incorporating new information on a static snapshot of the world (without changing generic or background knowledge). On the other hand, updating allows the world to have changed (and so adjusts the epistemic state to a possibly changed world). Thus, we can view updating as a successive process of changing the agent's epistemic state as new pieces of information arive. Genuine revision, on the other hand, collects the new pieces of information and executes simultaneous revision of the epistemic state.

In general, $(\kappa * \mathcal{R}) * \mathcal{S}$ - updating κ by $\mathcal{R}$ and the resulting state by $\mathcal{S}$ - differs from $\kappa * (\mathcal{R} \cup \mathcal{S})$ - revising $\kappa * \mathcal{R}$ by $\mathcal{S}$. Most obviously, this is the case for $\mathcal{R} = \{(B|A)\}$ and $\mathcal{S} = \{(\overline{B}|A)\}$. Here, updating κ by $\{(B|A)\}$ will succeed as well as updating the result with $\{(\overline{B}|A)\}$, which is information being contradictory to the first update information.

On the other hand, revising $\kappa * \{(B|A)\}$ by $\{(\overline{B}|A)\}$ fails because revising an epistemic state is only possible if the new piece of information is consistent with the currently held information. For instance, multiple observations shouldn't contradict one another.

We now want to demonstrate how the ASM framework offers a means to provide a very clear and precise distinction between update and revision operators in a knowledge processing system. Both operators are implemented by c-revisions of an epistemic state and a set of rules; the difference between update and revision lies in the exact specification of the parameters of the c-revison:

updateBlock $\equiv$
  stateBeforeUpdate := *currstate*;
  newRulesSinceUpdate := *new_information*;
  if *consistencyCheck*(*new_information*) = *false*
    then *output*("new information for update is inconsistent")
    else *currstate* := *cRevision*(*currstate*, *new_information*)

Thus, an update operation saves the actual current state (*currstate*) to *stateBeforeUpdate* and initializes the set of new rules (*newRulesSinceUpdate*) to be taken into account for changing the epistemic state to the the new information as given by the monitored function *new_information* : $\mathcal{P}(Sen_{\mathcal{O}})$. It then computes the new current state by a c-revision of the actual current state and the new information. A consistency check ensures that the information given to the c-revision is not contradictory.

reviseBlock $\equiv$
  newRulesSinceUpdate := *newRulesSinceUpdate* $\cup$ *new_information*;
  if *consistencyCheck*(*newRulesSinceUpdate*) = *false*
    then *output*("new information for revise is inconsistent")
    else *currstate* := *cRevision*(*stateBeforeUpdate*, *newRulesSinceUpdate*)

A revise operation, on the other hand, computes the new current state by a c-revision of the state being valid before the last update took place and the set of all new rules given for that last update and for all following single revision steps. Also here, a consistency check ensures that the information given to the c-revision is not contradictory.

Thus, in CondorASM$_{\mathcal{O}}$, update and genuine revision is realized by the rule

if *do* = *Revision*
  then if *rev_op*= *Update*
    then *updateBlock*
    else if *rev_op*= *Revision*
      then *reviseBlock*

where the monitored function *rev_op* : *RevisionOp* provides the type of revision operator to be applied.

Example 4. In Example 3, we illustrated the reasons for an agent with the given epistemic state κ to believe that kiwis have wings. Suppose now that the agent gets to know that this is false - kiwis do *not* possess wings - and we want the agent to adopt this new information. This can be done either by an update operation, in case the agent might take an evolutionary change of the world into consideration, or by a revision operation, in case it is certain that the birds' world has not changed. We will consider both possibilities, thereby illustrating the difference between both operations.

The updated epistemic state $\kappa_1^* = \kappa * \{(\overline{w}|k)\}$ is a c-revision of κ by $\{(\overline{w}|k)\}$. CondorASM$_{\mathcal{O}}$ computes this c-revision κ_1^* from κ by setting $\kappa_1^*(\omega) = \kappa(\omega) + 2$ for any ω with $\omega \models kw$ and setting $\kappa_1^*(\omega) = \kappa(\omega)$ otherwise. On the other hand, the (genuine) revision of κ by $(\overline{w}|k)$ is more complex, as a complete new inductive representation for the set $\mathcal{R}' = \{(f|b), (b|p), (\overline{f}|p), (w|b), (b|k)\} \cup \{(\overline{w}|k)\}$, i.e. a c-revision of *uniform* by $\mathcal{R}'$ has to be computed: $\kappa_2^* = \mathit{uniform} * \mathcal{R}'$.

While the revised state κ_2^*, by construction, still represents the five conditionals that have been known before (and, of course, the new conditional), it can be verified easily that the updated state κ_1^* only represents the four conditionals $(f|b)$, $(b|p)$, $(\overline{f}|p)$, and $(w|b)$, but it no longer satisfies $(b|k)$ because $\kappa_1^*(bk) = \kappa_1^*(\overline{b}k) = 1$ - since *birds* and *wings* have been plausibly related by the conditional $(w|b)$, the property of not having wings casts (reasonably) doubt on kiwis being birds. This illustrates that priorly stated, explicit knowledge is kept under revision, but might be given up under update.

4.6 Diagnosis

The process of diagnosing a particular case amounts to asking about the status of certain simple facts $D \subseteq \mathit{SimpleFact}_{\mathcal{U}} = \Sigma$ in an epistemic state Ψ under the condition that some particular factual knowledge $\mathcal{S}$ (so-called evidential knowledge) is given. Thus, an agent makes a diagnosis in the light of some given evidence by uttering his beliefs in the state obtained by adapting his current epistemic state by focussing on the given evidence.

Using the $belief_{\mathcal{O}}$ function as defined in Section 4.4 and again the concept of c-revision, diagnosis in CondorASM$_{\mathcal{O}}$ is realized by the rule

if $do = Diagnosis$
  then if $consistencyCheck(evidence) = false$
    then $output$("evidence for diagnosis is inconsistent")
    else let $focussedState = cRevision(currstate, evidence)$ **in**
      $believed\_sentences := belief_{\mathcal{O}}(focussedState, diagnoses)$

where the monitored functions $evidence : \mathcal{P}(Fact_{\mathcal{Q}})$ and $diagnoses : \mathcal{P}(SimpleFact_{\mathcal{U}})$ provide the factual evidence and a set of (unquantified) facts for which a degree of belief is to be determined.

Please note that the focussed epistemic state is only used for answering the particular diagnostic questions; specifically, the agent's current epistemic state ($currstate$) it not changed.

Example 5. Continuing Example 1, we might have the evidence for a penguin and want to ask for the diagnosis whether the penguin has wings. Here, CondorASM$_{\mathcal{O}}$ computes the degree of belief 1, i.e. this is a plausible diagnosis.

4.7 Hypothetical Reasoning

In diagnosis the agent's focus is set to a set of facts. The epistemic state reflecting this focus setting is obtained by an update operation on the agent's current epistemic state with respect to the evidential facts. On the other hand, in hypothetical reasoning *what-if* questions of the kind "*If these assumptions hold, what does that mean for some goals?*" are asked. In contrast to diagnosis, the assumptions considered may be not only factual evidence, but general pieces of knowledge. Also the goals may be not just simple facts, but complex sentences. Therefore, in hypothetical reasoning we model both assumptions and goals by general conditionals. The assumptions are expressed by quantified conditionals and the goals by unquantified ones whose degree of belief is asked for.

Apart from these differences, our general ASM framework allows us to nicely work out the structural similarities between diagnosis and hypothetical reasoning: Also the latter is achieved by querying an epistemic state that is obtained by an update operation on the agent's current epistemic state:

if $do = What\text{-}If\text{-}Analysis$
  then if $consistencyCheck(assumptions) = false$
    then $output$("assumptions for *what-if* analysis are inconsistent")
    else let $focussedState = cRevision(currstate, assumptions)$ **in**
      $believed\_sentences := belief_{\mathcal{O}}(focussedState, goals)$

where the assumptions used for hypotherical reasoning are being hold in the monitored function $assumptions : \mathcal{P}(Sen_{\mathcal{O}})$ and the sentences used as goals for which we ask for the degree of belief are being hold in the monitored function $goals : \mathcal{P}(Sen_{\mathcal{U}})$.

5 Conclusions and Further Work

In this paper, we presented CondorASM$_{\mathcal{O}}$, a refinement and implementation of CondorASM to qualitative conditionals equipped with the semantics of ordinal conditional functions. The ASM methodology allowed us to precisely describe sophisticated knowledge management tasks that are most prominently investigated in belief revision. For instance, we could elaborate the similarities and delicate differences between update and genuine revision operators.

CondorASM$_{\mathcal{O}}$ is implemented in XASM on a Linux PC [23]. Its ASM source code consists of approximately 30 pages, including a rudimentary interactive user interface and providing all functionalities described in this paper. Since XASM is no longer maintained and in order to exploit the rich functionalities of AsmL [3, 17], we plan to port CondorASM$_{\mathcal{O}}$ to AsmL for further development.

The Condor system as specified by CondorASM in [4] also provides the functionality for generating rules from data (*conditional knowledge discovery, CKD*), a process that is inverse to the inductive completion of knowledge given by a set of rules. So far we have elaborated and implemented CKD for the quantitive approach of probabilistic logic where probabilistic rules are generated from a full probability distribution [12]. For the case of qualitative logics with OCFs, the implementation of CKD from ranking functions is subject of our current work.

References

1. C.E. Alchourrón, P. Gärdenfors, and P. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.
2. M. Anlauff. XASM – An Extensible, Component-Based Abstract State Machines Language. In Y. Gurevich and P. Kutter and M. Odersky and L. Thiele, editor, *Abstract State Machines: Theory and Applications*, volume 1912 of *LNCS*, pages 69–90. Springer-Verlag, 2000.
3. AsmL webpage. `http://research.microsoft.com/foundations/asml/`, 2007.
4. C. Beierle and G. Kern-Isberner. Modelling conditional knowledge discovery and belief revision by Abstract State Machines. In E. Börger, A. Gargantini, and E. Riccobene, editors, *International Workshop on Abstract State Machines - ASM2003*, volume 2589 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003.
5. C. Beierle and G. Kern-Isberner. Footprints of conditionals. In D. Hutter and W. Stephan, editors, *Mechanizing Mathematical Reasoning*, volume 2605 of *Lecture Notes in Computer Science*, pages 99–119. Springer-Verlag, 2005.
6. S. Benferhat, D. Dubois, and H. Prade. Representing default rules in possibilistic logic. In *Proceedings 3th International Conference on Principles of Knowledge Representation and Reasoning KR'92*, pages 673–684, 1992.
7. S. Benferhat, A. Saffiotti, and P. Smets. Belief functions and default reasoning. *Artificial Intelligence*, 122:1–69, 2000.
8. C. Boutilier and M. Goldszmidt. Revision by conditional beliefs. In *Proceedings 11th National Conference on Artificial Intelligence (AAAI'93)*, pages 649–654, Washington, DC., 1993.

9. A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89:1–29, 1997.
10. B. DeFinetti. *Theory of Probability*, volume 1,2. John Wiley & Sons, 1974.
11. D. Dubois and H. Prade. Focusing vs. belief revision: A fundamental distinction when dealing with generic knowledge. In *Proceedings First International Joint Conference on Qualitative and Quantitative Practical Reasoning, ECSQARU-FAPR'97*, pages 96–107, Berlin Heidelberg New York, 1997. Springer.
12. J. Fisseler, G. Kern-Isberner, and C. Beierle. Learning uncertain rules with CONDORCKD. In *Proceedings 20th International FLAIRS Conference, FLAIRS'07*. AAAI Press, Menlo Park, California, 2007.
13. P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, Mass., 1988.
14. M. Goldszmidt, P. Morris, and J. Pearl. A maximum entropy approach to nonmonotonic reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):220–232, 1993.
15. M. Goldszmidt and J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence*, 84:57–112, 1996.
16. Y. Gurevich. Evolving Algebras 1993: Lipari Guide. In E. Börger, editor, *Specification and Validation Methods*, pages 9–36. Oxford University Press, 1995.
17. Y. Gurevich, B. Rossman, and W. Schulte. Semantic essence of AsmL. *Theoretical Computer Science*, 343(3):370–412, 2005.
18. H. Katsuno and A.O. Mendelzon. On the difference between updating a knowledge base and revising it. In *Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning, KR'91*, pages 387–394, San Mateo, Ca., 1991. Morgan Kaufmann.
19. G. Kern-Isberner. Characterizing the principle of minimum cross-entropy within a conditional-logical framework. *Artificial Intelligence*, 98:169–208, 1998.
20. G. Kern-Isberner. *Conditionals in nonmonotonic reasoning and belief revision*. Springer, Lecture Notes in Artificial Intelligence LNAI 2087, 2001.
21. G. Kern-Isberner. Handling conditionals adequately in uncertain reasoning. In *Proceedings European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'01*, pages 604–615. Springer LNAI 2143, 2001.
22. G. Kern-Isberner. A thorough axiomatization of a principle of conditional preservation in belief revision. *Annals of Mathematics and Artificial Intelligence*, 40(1-2):127–164, 2004.
23. M. Klockenkemper. Realisierung eines Systems zur Verarbeitung konditionalen Wissens mit ASMs. Diplomarbeit, FernUniversität in Hagen, 2005.
24. J.B. Paris. *The uncertain reasoner's companion – A mathematical perspective*. Cambridge University Press, 1994.
25. J.B. Paris and A. Vencovska. In defence of the maximum entropy inference process. *International Journal of Approximate Reasoning*, 17(1):77–103, 1997.
26. F.P. Ramsey. General propositions and causality. In R.B. Braithwaite, editor, *Foundations of Mathematics and other logical essays*, pages 237–257. Routledge and Kegan Paul, New York, 1950.
27. W. Spohn. Ordinal conditional functions: a dynamic theory of epistemic states. In W.L. Harper and B. Skyrms, editors, *Causation in Decision, Belief Change, and Statistics, II*, pages 105–134. Kluwer Academic Publishers, 1988.
28. R. Stärk, J. Schmid, and E. Börger. *Java and the Java Virtual Machine: Definition, Verification, Validation*. Springer-Verlag, 2001.
29. E. Weydert. Rankings we prefer: a rational minimal construction semantics for default reasoning. In *Proceedings DGNMR'01*, Potsdam, 2001.