

# Probabilistic Reasoning at Optimum Entropy with the MECoRE System\*

Marc Finthammer<sup>1</sup>, Christoph Beierle<sup>1</sup>, Benjamin Berger<sup>1</sup>, Gabriele Kern-Isberner<sup>2</sup>

<sup>1</sup>Dept. of Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany

<sup>2</sup>Dept. of Computer Science, TU Dortmund, 44221 Dortmund, Germany

{marc.finthammer, christoph.beierle}@fernuni-hagen.de, gabriele.kern-isberner@cs.uni-dortmund.de

## Abstract

Augmenting probabilities to conditional logic yields an expressive mechanism for representing uncertainty. The principle of optimum entropy allows one to reason in probabilistic logic in an information-theoretic optimal way by completing the given information as unbiasedly as possible. In this paper, we introduce the MECoRE system that realises the core functionalities for an intelligent agent reasoning at optimum entropy and that provides powerful mechanisms for belief management operations like revision, update, diagnosis, or hypothetical what-if-analysis.

## Introduction

When modelling an intelligent agent, elaborate knowledge representation and reasoning facilities are required. For instance, an agent performing medical diagnosis must be able to deal with pieces of knowledge expressed by rules such as “*If antibiotic A is effective against the observed bacteria (effect\_A), the patient will be healthy (outcome=healthy) with a probability of 80%.*” More formally, such uncertain probabilistic conditionals can be expressed by  $(\text{outcome}=\text{healthy}|\text{effect\_A})[0.8]$ . Certain knowledge as in “*If the observed bacteria is resistant to antibiotic A (resistance\_A), antibiotic A cannot be effective*” can be formalized by  $(\neg\text{effect\_A}|\text{resistance\_A})[1.0]$ . Furthermore, such an agent should be able to answer diagnostic questions in the presence of evidential facts like “*Given evidence for the resistance to antibiotic A, what are the patient’s healing chances if antibiotic B is given?*”

An agent capable of dealing with knowledge bases containing such general conditionals can be seen as an agent being able to take rules, pieces of evidence, queries, etc., from the environment and giving back sentences she believes to be true with a degree of certainty. Basically, these degrees of belief are inferred from the agent’s current epistemic state which is a representation of her cognitive state at the given time. When the agent is supposed to live in an uncertain and dynamic environment, she has to adapt her epistemic state

constantly to changes in the surrounding world and to react adequately to new demands (cf. (Darwiche and Pearl 1997), (Katsuno and Mendelzon 1991)).

In the following, we will deal with situations where basic knowledge in the form of quantitative knowledge is available where each conditional comes with a probability. Given an initial knowledge base  $KB$  of probabilistic conditionals, the epistemic state of a rational agent accepting  $KB$  can be expressed by a complete probability distribution  $P$  over the involved propositional variables. Any agent aiming to be as unbiased as possible should not presuppose any additional information apart from the explicitly given knowledge. Using an information-theoretic approach, this leads to the well-established concept of optimum entropy (Paris and Vencovska 1997; Kern-Isberner 1998).

In this paper, we introduce the MECoRE system that provides the core functionalities for an intelligent agent model as outlined above. MECoRE realises not only knowledge representation and inferencing under optimum entropy, but in particular, it provides powerful mechanisms for knowledge management operations like revision, update, diagnosis, or hypothetical what-if analysis.

We first provide a very brief introduction to probabilistic conditional logics, before the core functionalities of MECoRE are described. Afterwards, an overview of MECoRE is given, followed by a system walkthrough together with an application example.

## Probabilistic Conditional Logic in a Nutshell

We start with a propositional language  $\mathcal{L}$ , generated by a finite set  $\Sigma$  of (binary) atoms  $a, b, c, \dots$ . The formulas of  $\mathcal{L}$  will be denoted by uppercase Roman letters  $A, B, C, \dots$ . For conciseness of notation, we will omit the logical *and*-connector, writing  $AB$  instead of  $A \wedge B$ , and overlining formulas will indicate negation, i.e.  $\overline{A}$  means  $\neg A$ . Let  $\Omega$  denote the set of possible worlds over  $\mathcal{L}$ ;  $\Omega$  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  $\Sigma$ . For  $\omega \in \Omega$ ,  $\omega \models A$  means that the propositional formula  $A \in \mathcal{L}$  holds in the possible world  $\omega$ .

By introducing a new binary operator  $|$ , we obtain the set  $(\mathcal{L} | \mathcal{L}) = \{(B|A) \mid A, B \in \mathcal{L}\}$  of (unquantified) *conditionals* (or *rules*) over  $\mathcal{L}$ .  $(B|A)$  formalizes “*if A then B*” and establishes a plausible, probable, possible etc connec-

\*The research reported here was supported by the Deutsche Forschungsgemeinschaft (grants BE 1700/7-1 and KE 1413/2-1). Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

tion between the *antecedent*  $A$  and the *consequent*  $B$ . The set  $Sen_C$  contains all *probabilistic conditionals* (or *probabilistic rules*) of the form  $(B|A)[x]$  where  $x$  is a probability value  $x \in [0, 1]$ .

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 e.g. preferences, beliefs, assumptions of an intelligent agent. Basically, an epistemic state allows one to compare formulas or worlds with respect to plausibility, possibility, necessity, probability etc. In a quantitative framework, most appreciated representations of epistemic states are provided by *probability functions* (or *probability distributions*)  $P : \Omega \rightarrow [0, 1]$  with  $\sum_{\omega \in \Omega} P(\omega) = 1$ . Thus, in this setting, the set of *epistemic states* we will consider is  $EpState = \{P \mid P : \Omega \rightarrow [0, 1] \text{ is a probability function}\}$ . The probability of a formula  $A \in \mathcal{L}$  is given by  $P(A) = \sum_{\omega \models A} P(\omega)$ , and the probability of a conditional  $(B|A) \in (\mathcal{L} \mid \mathcal{L})$  with  $P(A) > 0$  is defined as  $P(B|A) = P(AB)/P(A)$ , the corresponding conditional probability. Conditionals are interpreted via conditional probability. So the satisfaction relation  $\models_C \subseteq EpState \times Sen_C$  of probabilistic conditional logic is defined by  $P \models_C (B|A)[x]$  iff  $P(B|A) = x$ .

## Core Functions of the MECoRe-System

### Initialization

In the beginning, a prior epistemic state has to be built up on the basis of which the agent can start her computations. If no knowledge at all is at hand, simply the uniform epistemic state is taken to initialize the system. In our probabilistic setting, this corresponds to the uniform distribution where each possible world is assigned the same probability. If, however, a set of probabilistic rules 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, we assume an inductive representation method to establish the desired connection between sets of sentences and epistemic states. Whereas generally, a set  $R$  of sentences allows a (possibly large) set of models (or epistemic states), in an inductive formalism we have a function *inductive* :  $\mathcal{P}(Sen_C) \rightarrow EpState$  such that *inductive*( $R$ ) selects a unique, “best” epistemic state from all those states satisfying  $R$ .

In the probabilistic framework, the *principle of maximum entropy* associates to a set  $R$  of probabilistic conditionals the unique distribution  $P^* = MaxEnt(R)$  that satisfies all conditionals in  $R$  and has maximal entropy, i.e.,  $MaxEnt(R)$  is the *unique* solution to the maximization problem

$$\arg \max_{P' \models R} H(P') = - \sum_{\omega} P'(\omega) \log P'(\omega) \quad (1)$$

The rationale behind this is that  $MaxEnt(R)$  represents the knowledge given by  $R$  most faithfully, i.e. without adding information unnecessarily (cf. (Paris and Vencovska 1997; Kern-Isberner 1998)). We will illustrate the maximum entropy method by a small example.

**Example 1** Consider the three propositional variables  $s$  - being a student,  $y$  - being young, and  $u$  - being unmarried. Students and unmarried people are mostly young. This commonsense knowledge an agent may have can be expressed probabilistically e.g. by the set  $R = \{(y|s)[0.8], (y|u)[0.7]\}$  of conditionals. The *MaxEnt-representation*  $P^* = MaxEnt(R)$  computed by MECoRe is:

$\omega$	$P^*(\omega)$	$\omega$	$P^*(\omega)$	$\omega$	$P^*(\omega)$	$\omega$	$P^*(\omega)$
$syu$	0.1950	$sy\bar{u}$	0.1758	$\bar{s}yu$	0.0408	$\bar{s}y\bar{u}$	0.0519
$\bar{s}yu$	0.1528	$\bar{s}y\bar{u}$	0.1378	$\bar{s}\bar{y}u$	0.1081	$\bar{s}\bar{y}\bar{u}$	0.1378

### Querying an Epistemic State

Querying an agent about her beliefs amounts to pose a set of unquantified sentences and asking for the corresponding degrees of belief with respect to her current epistemic state.

**Example 2** Suppose the current epistemic state is  $currState = MaxEnt(R)$  from Ex. 1, and our question is “What is the probability that unmarried students are young?”, i.e. the set of queries is  $\{(y|su)\}$ . MECoRe returns  $\{(y|su)[0.8270]\}$ , that is, unmarried students are supposed to be young with probability 0.8270.

### New Information and Belief Change

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 (Alchourrón, Gärdenfors, and Makinson 1985) as a set of postulates outlining appropriate revision mechanisms in a propositional logical environment. This framework has been widened by Darwiche and Pearl (Darwiche and Pearl 1997) 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 (Kern-Isberner 2001). The crucial meaning of conditionals as *revision policies* for belief revision processes is made clear by the so-called *Ramsey test*, according to which a conditional  $(B|A)$  is accepted in an epistemic state  $\Psi$ , iff revising  $\Psi$  by  $A$  yields belief in  $B$ :  $\Psi \models (B|A)$  iff  $\Psi * A \models B$  where  $*$  is a belief revision operator (see e.g. (Gärdenfors 1988)).

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 (Gärdenfors 1988). On a more sophisticated level, however, one distinguishes between different kinds of belief change. Here, (*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 (Katsuno and Mendelzon 1991). Further belief change operators are *expansion*, *focusing*, *contraction*, and *erasure* (cf. (Gärdenfors 1988; Dubois and Prade 1997; Katsuno and Mendelzon 1991)). In the following, we will use the general approach to belief change developed in (Kern-Isberner 2001) where belief change is considered in a very general and advanced form: Epistemic states are revised by sets of conditionals – this exceeds the classical AGM-theory by far which only deals with sets of propositional beliefs. Due to space restrictions, we will concentrate

on the core functionality of updating in this paper. In (Kern-Isberner 2008) it is shown how belief revision can be based on such a change operator.

In the probabilistic framework, a powerful tool to update probability distributions by sets of probabilistic conditionals is provided by the *principle of minimum cross-entropy* which generalizes the principle of maximum entropy in the sense of (1): Given a (prior) distribution  $P$  and a set  $R$  of probabilistic conditionals, the *MinCent-distribution*  $P^* = \text{MinCent}(P, R)$  is the *unique* distribution that satisfies all constraints in  $R$  and has minimal cross-entropy  $H_{ce}$  with respect to  $P$ , i.e.  $P^*$  solves the minimization problem

$$\arg \min_{P' \models R} H_{ce}(P', P) = \sum_{\omega} P'(\omega) \log \frac{P'(\omega)}{P(\omega)} \quad (2)$$

If  $R$  is basically compatible with  $P$  (i.e.  $P$ -consistent, cf. (Kern-Isberner 2001)), then  $P^*$  is guaranteed to exist (for further information and lots of examples, see (Csiszár 1975; Paris and Vencovská 1992; Kern-Isberner 2001)). The cross-entropy between two distributions can be taken as a directed (i.e. asymmetric) information distance (Shore 1986) between these two distributions. Following the principle of minimum cross-entropy means to modify the prior epistemic state  $P$  in such a way as to obtain a new distribution  $P^*$  which satisfies all conditionals in  $R$  and is as close to  $P$  as possible. So, the *MinCent*-principle yields a probabilistic belief update operator, associating to each probability distribution  $P$  and each  $P$ -consistent set  $R$  of probabilistic conditionals a revised distribution  $P^* = \text{MinCent}(P, R)$  in which  $R$  holds.

**Example 3** Suppose that some time later, the relationships in the population from Example 1 between students and young people have changed, so that students are young with a probability of 0.9. In order to incorporate this new knowledge, the agent applies an updating operation to modify  $P^*$  appropriately. The result  $P^{**} = \text{MinCent}(P^*, \{(y|s)[0.9]\})$  as determined by MECoRe is:

$\omega$	$P^{**}(\omega)$	$\omega$	$P^{**}(\omega)$	$\omega$	$P^{**}(\omega)$	$\omega$	$P^{**}(\omega)$
$syu$	0.2151	$sy\bar{u}$	0.1939	$s\bar{y}u$	0.0200	$s\bar{y}\bar{u}$	0.0255
$\bar{s}yu$	0.1554	$\bar{s}y\bar{u}$	0.1401	$\bar{s}\bar{y}u$	0.1099	$\bar{s}\bar{y}\bar{u}$	0.1401

It is easily checked that indeed,  $P^{**}(y|s) = 0.9$  (only approximately, due to rounding errors).

## Diagnosis

Diagnosing a given case is one of the most common operations in knowledge based systems. Given some case-specific evidence  $E$  (formally, a set of quantified facts), diagnosis assigns degrees of belief to the atomic propositions  $D$  to be *diagnosed* (formally,  $D$  is a set of unquantified atomic propositions). Thus, making a diagnosis in the light of some given evidence corresponds to determine what is believed in the state obtained by focusing the current state  $P$  on the given evidence, i.e. querying the epistemic state  $\text{MinCent}(P, E)$  with respect to  $D$ . Thus, here focusing corresponds to conditioning  $P$  with respect to the given evidence  $E$ .

**Example 4** Let  $\text{currState} = P^*$  from Ex. 1. If there is now certain evidence for being a student and being unmarried

– i.e.  $E = \{su[1]\}$  – and we ask for the degree of belief of being young – i.e.  $D = \{y\}$  –, MECoRe computes  $\{y[0.8270]\}$ . Thus, if there is certain evidence for being an unmarried student, then the degree of belief for being young is 0.8270.

## What-If-Analysis: Hypothetical Reasoning

Hypothetical reasoning asks for the degree of belief of complex relationships (goals) under some hypothetical assumptions. This is useful, e.g., to exploit in advance the benefits of some expensive or intricate medical investigations. Note that whereas in the diagnostic case both evidence  $E$  and diagnoses  $D$  are just simple propositions, in hypothetical reasoning both the *assumptions*  $A$  (formally, a set of quantified conditionals) as well as the *goals*  $G$  (formally, a set of unquantified conditionals) may be sets of full conditionals. However, since its underlying powerful *MinCent*-update operator can modify epistemic states by arbitrary sets of conditionals, MECoRe can handle hypothetical what-if-analysis structurally analogously to the diagnostic case, i.e. by querying the epistemic state  $\text{focussed\_state} = \text{MinCent}(P, A)$  with respect to  $G$  where  $P$  is the current epistemic state. Since this is hypothetical reasoning, the agent's current epistemic state remains unchanged.

**Example 5** Given  $\text{currState} = P^*$  from Ex. 1 as present epistemic state, a hypothetical reasoning question is given by: “What would be the probability of being young under the condition of being unmarried – i.e.  $G = \{(y|u)\}$  –, provided that the probability of a student being young changed to 0.9 – i.e.  $A = \{(y|s)[0.9]\}$ ?” MECoRe's answer is  $\{(y|u)[0.7404]\}$  which corresponds to the probability given by  $P^{**}$  from Ex. 3.

## The MECoRe system

Besides providing the core functionalities needed for probabilistic reasoning at optimum entropy, the main objective of MECoRe is to support advanced belief management operations like revision, update, diagnosis, or what-if-analysis in a most flexible and easily extendible way. MECoRe is implemented in Java. In its current version, it uses a straightforward, direct implementation of a well-known *MinCent* algorithm and provides a very powerful and flexible interface.

## Computation of *MinCent*

MECoRe computes the distribution  $P^* = \text{MinCent}(P, R)$  in an iterative way. A detailed description of the used algorithm can be found in (Csiszár 1975). In principle, the algorithm iterates over all rules in  $R$  in a cyclical order. In each iteration step, only one rule is considered and the current distribution (starting with  $P$ ) is appropriately adjusted to satisfy the considered rule. It can be shown that this iterative process converges to  $P^*$ .

The current implementation of the algorithm in MECoRe works on an explicit representation of the whole probability distribution. Although this representation has to cope with the exponential size in the number of variables, it is still efficient enough to compute knowledge bases with 20 variables

in about 30 seconds. Since the algorithm is implemented independently from the actual representation of the distribution, the MECORE system can easily be extended by a more sophisticated and efficient kind of representation of the distribution, e. g. by a junction tree (Teh and Welling 2003).

## The User Interface

The present version of MECORE can be controlled by a text command interface or by scripts, i. e. text files that allow the batch processing of command sequences. These scripts and the text interface use a programming language-like syntax that allows to define, manipulate and display variables, propositions, rule sets and epistemic states. The following example shows a way to generate an epistemic state using the initialize and update operators:

```
//define a set of rules
kb := ((y|s)[0.8], (y|u)[0.7]);
// initialize an epistemic state with these rules
currState := epstate().initialize(kb);
//query and output current belief in the conditional (y|su)
currState.query((y|su));
//update the epistemic state currState by (y|s)[0.9]
currState.update((y|s)[0.9]);
```

Hence, one is able to use both previously defined rule sets and rules that are entered just when they are needed, and combinations of both. The ability to manipulate rule sets, to automate sequences of updates and revisions, and to output selected (intermediate) results for comparing, yields a very expressive command language. This command language is a powerful tool for experimenting and testing with different setups. All core functions of the MECORE system are also accessible through a software interface (in terms of a Java API). So MECORE can easily be extended by a GUI or be integrated into another software application.

## Related Work

There are many systems performing inferences in probabilistic networks, especially in Bayesian networks. One system built upon network techniques to implement reasoning at optimum entropy is the expert system shell SPIRIT (Rödder, Reucher, and Kulmann 2006). Graph based methods are known to feature a very efficient representation of probability distributions via junction trees and hypergraphs, while MECORE works on a model based representation of probabilities. This is clearly inefficient, but efficiency is not the point here. The aim of the MECORE project is to implement subjective probabilistic reasoning, as it could be performed by agents, making various belief operations possible. In particular, it allows changing of beliefs in a very flexible way by taking new, complex information into account. This is not possible with graph based systems for probabilistic inference, as no efficient methods of restructuring probabilistic networks have been developed to date.

## System Walkthrough and Example Application

This example will illustrate how incomplete, uncertain knowledge can be expressed by a probabilistic knowledge

base. It will also demonstrate how new knowledge can be inferred and how hypothetical reasoning can be performed.

This (fictitious) example from the medical domain discusses the general treatment of a patient who suffers from a perilous bacterial infection. The infection will probably cause permanent neurological damage or even death if it is not treated appropriately. There are two antibiotics available that might be capable of ending the infection, provided that the bacterial is not resistant to the specific antibiotic. It must also be considered that each antibiotic might cause a life-threatening allergic reaction that could be hard to survive for the already weakened patient. The resistance of the bacterial to a specific antibiotic can be tested, but each test is very time-consuming.

## Building Up the Knowledge Base

The construction of the knowledge base starts with the definition of some binary variables that describe aspects concerning antibiotic A:

med\_A: The patient is treated with antibiotic A.  
effect\_A: Antibiotic A is effective against the bacteria.  
allergic\_A: The patient is allergic to antibiotic A.  
resistance\_A: The bacteria are resistant to antibiotic A.  
posResTest\_A: The test result suggests a resistance to antibiotic A.

Five corresponding variables concerning antibiotic B are added to the knowledge base as well. A three-valued variable outcome describes the three possible outcomes of the treatment:

outcome=healthy: The infection is treated successfully and the patient is healthy again.  
outcome=impaired: The patient overcomes the infection but suffers a permanent damage to the nervous system.  
outcome=dead: The infection is not treated effectively and the patient dies.

The available knowledge summarizing the previously made experiences about the infection and the two antibiotics is modeled by the following probabilistic rules:

```
R1 : (¬effect_A|¬med_A ∨ resistance_A)[1.00]
R2 : (¬effect_B|¬med_B ∨ resistance_B)[1.00]
R3 : (effect_A ⇔ med_A|¬resistance_A)[1.00]
R4 : (effect_B ⇔ med_B|¬resistance_B)[1.00]
R5 : (allergic_A)[0.10]
R6 : (allergic_B)[0.20]
R7 : (resistance_A)[0.01]
R8 : (resistance_B)[0.09]
R9 : (med_A ∧ med_B)[0.00001]
R10: (outcome=dead|¬med_A ∧ ¬med_B)[0.10]
R11: (outcome=healthy|¬med_A ∧ ¬med_B)[0.10]
R12: (posResTest_A|resistance_A)[0.97]
R13: (¬posResTest_A|¬resistance_A)[0.99]
R14: (posResTest_B|resistance_B)[0.90]
R15: (¬posResTest_B|¬resistance_B)[0.80]
R16: (outcome=dead|med_A ∧ allergic_A)[0.99]
R17: (outcome=dead|med_B ∧ allergic_B)[0.40]
R18: (outcome=healthy|effect_A)[0.8]
R19: (outcome=healthy|effect_B)[0.7]
R20: (allergic_A|med_A)[0.10]
R21: (outcome=dead|effect_B)[0.09]
R22: (outcome=healthy|med_B ∧ allergic_B)[0.001]
```

The first four rules express very obvious correlations between the variables:  $R_1$  and  $R_2$  say that if a certain antibiotic is not administered or the bacteria are resistant to it, then this antibiotic has no effect.  $R_3$  and  $R_4$  assure that if the bacteria are not resistant to a certain antibiotic, then this antibiotic is effective if – and only if – it is administered. The facts  $R_5$  to  $R_9$  integrate statistical information available for antibiotic A and antibiotic B, i.e. some a priori probabilities, into the knowledge base: Antibiotic B is twice as likely as antibiotic A to cause an allergic reaction ( $R_5, R_6$ ); and the resistance to antibiotic B is nine times higher compared to antibiotic A ( $R_7, R_8$ ). It has occurred very rarely that somebody administers both antibiotics to the patient ( $R_9$ ).  $R_{10}$  and  $R_{11}$  model the prognosis for the patient if no antibiotic is administered. The result of a resistance-test, testing the resistance of the bacteria to an antibiotic, always includes some error, but the test regarding antibiotic A is very reliable ( $R_{12}, R_{13}$ ); whereas the test concerning antibiotic B has a somewhat lower sensitivity ( $R_{14}$ ) and a considerably lower specificity ( $R_{15}$ ).

The rules  $R_{16}$  to  $R_{19}$  express special knowledge about antibiotic A and antibiotic B, respectively: The allergic reaction caused by antibiotic A is most likely lethal ( $R_{16}$ ), whereas the chance of surviving an allergy to antibiotic B is more likely than to die of it ( $R_{17}$ ). If antibiotic A is effective, then the patient's has a good chance to become healthy again ( $R_{18}$ ), whereas the effectiveness of antibiotic B is somewhat lower ( $R_{19}$ ). The following knowledge is available for antibiotic A only:  $R_{20}$  makes clear that the a priori probability of an allergy to antibiotic A (expressed by  $R_5$  with equal probability) is not affected by the administration of antibiotic A. There is also some exclusive knowledge about antibiotic B: If antibiotic B is effective, there still remains some risk to die of the infection ( $R_{21}$ ). If the patient survives an allergic reaction caused by antibiotic B, it is very unlikely that he will become healthy again ( $R_{22}$ ).

The rules  $R_1$  to  $R_{22}$ , i.e. the rules which make up the knowledge base, are assigned to a named rule set `medKB` to make them directly accessible for knowledge processing operations.

## Knowledge Processing

**Initialization** Once all knowledge rules have been defined, the computation of an epistemic state incorporating this knowledge can be started by the following command:

```
(1) currState := epstate.initialize(medKB);
```

The calculated epistemic state `currState` represents the (incomplete) knowledge expressed by `medKB` inductively completed in an entropy-optimal way.

A closer look at `medKB` reveals that some additional rules can be logically deduced from the existing rules since they hold in all models satisfying `medKB`. For instance, a literal of the three-valued variable outcome makes up the conclusion of several rules. Hence, two rules with identical premise and an outcome literal as conclusion directly imply a corresponding third rule, e.g.  $R_{10}$  and  $R_{11}$  imply  $(\text{outcome} = \text{impaired} | \neg \text{med.A} \wedge \neg \text{med.B})[0.8]$ . Likewise, the knowledge that is available for antibiotic B in rules  $R_{19}$  and  $R_{21}$  implies  $(\text{outcome} = \text{impaired} | \text{effect.B})[0.21]$ .

Appropriate queries to MECoRE in `currState` yield these expected probabilities since reasoning at optimum entropy is compatible with classical probabilistic consequences.

**Query** Let us now formulate some questions that we want to be answered, i.e. inferred from `currState`. We want to know the patient's chances in each case of treatment, i.e. for each of the four possible options of medical administration: no antibiotic, antibiotic A only, antibiotic B only, both antibiotics. These questions are expressed by twelve query formulas (i.e. conditionals), constructed as follows:

```
(outcome = healthy |  $\neg \text{med.A} \wedge \neg \text{med.B}$ ),
(outcome = impaired |  $\neg \text{med.A} \wedge \neg \text{med.B}$ ),
(outcome = dead |  $\neg \text{med.A} \wedge \neg \text{med.B}$ ),
(outcome = healthy |  $\text{med.A} \wedge \neg \text{med.B}$ ),
(outcome = impaired |  $\text{med.A} \wedge \neg \text{med.B}$ ),
...
(outcome = impaired |  $\text{med.A} \wedge \text{med.B}$ ),
(outcome = dead |  $\text{med.A} \wedge \text{med.B}$ )
```

We assign these twelve queries to a query set named `medQueries`. Query sets play an important role in belief processing, as they serve to make relevant beliefs from the complex epistemic state explicit and usable for the problem under consideration. An appropriate query set is also helpful to illustrate the effects of all kinds of belief revision.

The queries in `medQueries` are evaluated by the following command:

```
(2) currState.query(medQueries);
```

The MECoRE system calculates the following probabilities:

	healthy	impaired	dead
no antibiotic	0.10	0.80	0.10
antibiotic A only	0.79	0.06	0.15
antibiotic B only	0.65	0.23	0.12
both antibiotics	0.94	0.02	0.04

These results clearly suggest that the combined administration of both antibiotics would be the best treatment. It offers a high chance of healing accompanied by a minimal risk of permanent neurological damage or death.

However, a closer look at the knowledge base reveals that it contains no knowledge about a possible drug interaction. Asking for the degree of belief for the conditional

$C_{\text{int}} : (\text{dead} | \text{med.A} \wedge \text{med.B} \wedge \neg \text{allergic.A} \wedge \neg \text{allergic.B})$  by performing the query

```
(3) currState.query(Cint);
```

yields the inferred drug interaction probability 0.01.

**Incorporation of New Knowledge** Suppose the doctor learns to know from an outside source that there is a severe risk (0.25) of a deadly drug interaction between both antibiotics. Executing

```
(4) currState.update(medKB, Cint[0.25]);
```

incorporates this new knowledge into the current epistemic state as if it had been available already in `medKB`. In fact, this kind of belief change is a *genuine revision* (cf. (Kern-Isberner 2008)) which in MECoRE can also be more easily expressed by

```
(4') currState.revise(Cint[0.25]);
```

Now, asking the `medQueries` again, the probabilities have changed considerably:

```
(5) currState.query(medQueries);
```

	healthy	impaired	dead
no antibiotic	0.10	0.80	0.10
antibiotic A only	0.79	0.06	0.15
antibiotic B only	0.65	0.23	0.12
both antibiotics	0.70	0.02	0.28

With the knowledge about a deadly drug interaction, the probabilities show that the administration of antibiotic A maximizes the patient's chance to become healthy again.

**What-If-Analysis** It has to be noticed that in the epistemic state `currState` no resistance-tests have been performed, i. e. for neither of the antibiotics any resistance-test results are available.

A what-if-analysis can be used to analyze what changes would occur if a negative resistance-test result concerning antibiotic B was known. That is, could this test result make antibiotic B the better choice for treatment? Such a what-if-analysis is accomplished by the following command:

```
(6) currState.whatif((-posResTest.B)[1.0],medQueries);
```

The what-if-analysis delivers this results:

	healthy	impaired	dead
no antibiotic	0.10	0.80	0.10
antibiotic A only	0.79	0.06	0.15
antibiotic B only	0.69	0.21	0.10
both antibiotics	0.76	0.02	0.22

The probabilities show that even a negative resistance-B-test would not change the general decision to administer antibiotic A. This result is, amongst others, caused by the low resistance-B-test specificity.

Another what-if-analysis can reveal the effects a positive resistance-A-test would induce:

```
(7) currState.whatif((posResTest.A)[1.0],medQueries);
```

	healthy	impaired	dead
no antibiotic	0.10	0.80	0.10
antibiotic A only	0.43	0.15	0.42
antibiotic B only	0.65	0.23	0.12
both antibiotics	0.32	0.05	0.63

This shows that a test-result suggesting the resistance to antibiotic A would change the situation: In this case, a treatment with antibiotic B becomes the only that offers a realistic healing-chance. This is not surprising, because a resistance-test result concerning antibiotic A is very reliable. So it is clearly advisable to perform the time-consuming resistance-A-test.

In case of a positive resistance-A-test result, would it also be helpful to test the resistance to antibiotic B? That is, could an additional positive resistance-B-test change the decision to administer antibiotic B?

```
(8) currState.whatif(((posResTest.A)[1.0],
(posResTest.B)[1.0]),medQueries);
```

	healthy	impaired	dead
no antibiotic	0.10	0.80	0.10
antibiotic A only	0.43	0.15	0.42
antibiotic B only	0.54	0.26	0.20
both antibiotics	0.20	0.04	0.76

The what-if-analysis shows that even a positive resistance-B-test would not change the decision to administer antibiotic B. So it is not helpful to perform a resistance-B-test in any situation, since its result would never change the decision that had been made without knowing the test-result.

## Conclusions and Further Work

The main objective when developing MECoRe was to provide the core functionalities needed for probabilistic reasoning at optimum entropy and to support advanced belief management operations in a most flexible and easily extendible way. After having reached these goals, our current work includes using factorizations of the probability distributions for an optimized internal, graph-based representation, to complement MECoRe's expressive text and script based interface by a graphical user interface, and to study further example applications.

## References

- Alchourrón, C.; Gärdenfors, P.; and Makinson, P. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50(2):510–530.
- Csiszár, I. 1975. I-divergence geometry of probability distributions and minimization problems. *Ann. Prob.* 3:146–158.
- Darwiche, A., and Pearl, J. 1997. On the logic of iterated belief revision. *Artificial Intelligence* 89:1–29.
- Dubois, D., and Prade, H. 1997. Focusing vs. belief revision: A fundamental distinction when dealing with generic knowledge. In *Proc. ECSQARU-FAPR'97*, 96–107. Springer.
- Gärdenfors, P. 1988. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press.
- Katsuno, H., and Mendelzon, A. 1991. On the difference between updating a knowledge base and revising it. In *Proc. KR'91*, 387–394. Morgan Kaufmann.
- Kern-Isberner, G. 1998. Characterizing the principle of minimum cross-entropy within a conditional-logical framework. *Artificial Intelligence* 98:169–208.
- Kern-Isberner, G. 2001. *Conditionals in nonmonotonic reasoning and belief revision*. Springer, LNAI 2087.
- Kern-Isberner, G. 2008. Linking iterated belief change operations to nonmonotonic reasoning. In Brewka, G., and Lang, J., eds., *Proc. KR'2008*, 166–176. AAAI Press.
- Paris, J., and Vencovská, A. 1992. A method for updating that justifies minimum cross entropy. *International Journal of Approximate Reasoning* 7:1–18.
- Paris, J., and Vencovska, A. 1997. In defence of the maximum entropy inference process. *International Journal of Approximate Reasoning* 17(1):77–103.
- Rödter, W.; Reucher, E.; and Kulmann, F. 2006. Features of the expert-system-shell SPIRIT. *Logic Journal of the IGPL* 14(3):483–500.
- Shore, J. 1986. Relative entropy, probabilistic inference and AI. In Kanal, L., and Lemmer, J., eds., *Uncertainty in Artificial Intelligence*. Amsterdam: North-Holland. 211–215.
- Teh, Y. W., and Welling, M. 2003. On improving the efficiency of the iterative proportional fitting procedure. In *Proc. AISTATS'2003*. Society for Artificial Intelligence and Statistics.