4th Workshop on
Dynamics of Knowledge and Belief
(DKB-2013)

at the 36th Annual German Conference on Artificial Intelligence, KI-2013
Koblenz, Germany, September 17, 2013

Proceedings

Christoph Beierle, Gabriele Kern-Isberner (Eds.)
4th Workshop on
Dynamics of Knowledge
and Belief
(DKB-2013)

Workshop at the
36th Annual German Conference
on Artificial Intelligence, KI-2013
Koblenz, Germany, September 17, 2013

Proceedings
Workshop Organization

Special Interest Group on Knowledge Representation and Reasoning of the Gesellschaft für Informatik

(GI-Fachgruppe “Wissensrepräsentation und Schließen”)

in cooperation with:

DFG Priority Program on New Frameworks of Rationality (SPP 1516)

Workshop Co-Chairs

Gabriele Kern-Isberner  Technische Universität Dortmund, Germany
Christoph Beierle  FernUniversität in Hagen, Germany

Program Committee

Franz Baader  TU Dresden, Germany
Piero Andrea Bonatti  University of Napoli, Italy
Gerd Brewka  Universität Leipzig, Germany
François Bry  LMU München, Germany
Thomas Eiter  TU Wien, Austria
Eduardo Fermé  Universidade da Madeira, Portugal
Andreas Herzig  Université Paul Sabatier, Toulouse, France
Kristian Kersting  Fraunhofer IAIS, Universität Bonn, Germany
Markus Knauff  Univ. Giessen, Germany
Gerhard Lakemeyer  RWTH Aachen, Germany
Carsten Lutz  Univ. Bremen, Germany
Thomas Meyer  CSIR Meraka & University of KwaZulu-Natal, ZA
Marco Ragni  Univ. Freiburg, Germany
Sebastian Rudolph  TU Dresden, Germany
Klaus Dieter Schewe  Software Competence Center Hagenberg (SCCH), Austria
Gerhard Schurz  Univ. Dsseldorf, Germany
Matthias Thimm  TU Dortmund, Germany
Hans Tompits  TU Wien, Austria
Stefan Woltran  TU Wien, Austria

Additional Reviewers

Richard Booth
Stefan Rümmele
Preface

Information for real life AI applications is usually pervaded by uncertainty and subject to change, and thus demands for non-classical processing. A rich palette of methods for uncertain reasoning have been developed in the field of knowledge representation showing its many facets - qualitative vs. quantitative reasoning, argumentation and negotiation in multi-agent systems, causal reasoning for action and planning, as well as nonmonotonicity and belief revision have become very active fields of research, among many others.

This volume contains the contributions that were presented at the Workshop on Dynamics of Knowledge and Belief (DKB-2013) on September 17, 2013, in Koblenz, Germany, co-located with the 36th Annual German Conference on AI (KI-2013). It was organized by the Special Interest Group on Knowledge Representation and Reasoning of the Gesellschaft für Informatik (GI-Fachgruppe Wissensrepräsentation und Schließen), in cooperation with the DFG Priority Program on New Frameworks of Rationality (DFG Schwerpunkt-Programm SPP 1516). This was the 4th Workshop on Dynamics of Knowledge and Belief, following previous workshops at KI-2007 in Osnabrück, at KI-2009 in Paderborn, and at KI-2011 in Berlin.

The aim of this series of workshops is to address recent challenges and to present novel approaches to uncertain reasoning and belief change in their broad senses, and in particular provide an interdisciplinary forum for research work linking different paradigms of reasoning. The workshops have also been fostering cross-fertilization between the area of knowledge representation, on the one hand, and other disciplines such as philosophy and psychology, on the other hand.

Igor Douven gave an invited talk at the workshop. In his presentation Conditionals and Inferential Connections, he observes that many have the intuition that for a conditional to be true, there must exist some kind of connection between its antecedent and its consequent. This observation is investigated experimentally, and the data obtained from this experiment are better explained by a semantics for conditionals that makes inferential connectedness a truth condition than by any of the currently more popular semantics for conditionals.

In his paper Dynamic Preference Aggregation under Preference Changes, Matthias Thimm considers the issue of update in settings for preference aggregation under preference changes. His analysis shows that even for some simple aggregation rules, i.e. for the plurality and the Borda rule, the dynamic updating of an aggregated preference order can be handled more efficiently than recomputing the order from scratch when changes have to be made.

In Characteristics of Multiple Viewpoints in Abstract Argumentation, Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran present a formal basis for examining extension-based semantics of argumentation framework. They provide a number of characterization theorems which guarantee the existence of argumentation frameworks whose set of extensions satisfy specific conditions,
and derive preliminary complexity results for decision problems that require such characterizations.

Patrick Krümpelmann and Gabriele Kern-Isberner study secrecy from the point of view of an autonomous knowledge-based and resource-bound agent in Secrecy preserving BDI Agents based on Answerset Programming. They develop a general epistemic agent model with its secrecy relevant components and illustrate it by using a BDI-based agent model and answer set programming for knowledge representation.

Similar to Bayesian networks, OCF-networks combine structural information encoded in a directed graph with qualitative information expressed by ranking degrees of formulas. In OCF-Networks with Missing Values, Gabriele Kern-Isberner and Christian Eichhorn apply inductive reasoning methods to fill up missing values in local conditional tables, allowing the user to specify knowledge for such OCF-networks while leaving the technical details to an inference engine.

In a session on modeling and reasoning with quantitative methods, three papers investigating aspects of probabilistic approaches were presented. In their paper An Overview of Algorithmic Approaches to Compute Optimum Entropy Distributions in the Expert System Shell MECore, Nico Potyka, David Marenke, and Engelbert Mittermeier work with the MECore system that employs the principle of maximum entropy. They report on ongoing further developments of MECore’s algorithms for computing optimum entropy distributions, pointing out their benefits as well as possible limitations and pitfalls.

In Ampliative Inference Under Varied Entropy Levels, Paul Thorn and Gerhard Schurz investigate inference from conditional assertions that are interpreted as expressing high conditional probabilities using the four well known systems O, P, Z, and QC. They present data charting the performance of the four systems in reasoning about probability distributions with various entropy levels, allowing for a more inclusive assessment of the reliability and robustness of these systems.

Adaptive knowledge modeling is an approach to extend the abilities of the Object-Oriented World Model for allowing it to cope with open environments in which unforeseen entities can occur. The paper Quantitative Measures for Adaptive Object-Oriented World Modeling by Achim Kuwertz and Jürgen Beyerer extends adaptive knowledge management that was introduced in previous work. Further measures for identifying points of change in knowledge models are proposed and an approach for model adaptation is presented.

We would like to thank all Program Committee members as well as the additional external reviewers for detailed and high-quality reviews for all submitted papers. Many thanks also to the organizers of KI-2013 for hosting the workshop at the KI-2013 conference. Finally, we would like to thank the Gesellschaft für Informatik, the DFG Priority Program on New Frameworks of Rationality (SPP 1516), the FernUniversität in Hagen, and the TU Dortmund for supporting this workshop.

Hagen and Dortmund, September 2013

Christoph Beierle and Gabriele Kern-Isberner
Contents

Workshop Organization ......................................................... ii
Preface ................................................................................ iii
  Christoph Beierle, Gabriele Kern-Isberner

Invited Talk

Conditionals and Inferential Connections ................................. 1
  Igor Douven

Preferences, Argumentation and Agents

Dynamic Preference Aggregation under Preference Changes .......... 2
  Matthias Thimm
Characteristics of Multiple Viewpoints in Abstract Argumentation .... 16
  Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, Stefan Woltran
Secrecy preserving BDI Agents based on AnswerSet Programming ...... 31
  Patrick Krümpelmann, Gabriele Kern-Isberner

Uncertain Reasoning and Qualitative Conditionals

OCF-Networks with Missing Values ......................................... 46
  Gabriele Kern-Isberner and Christian Eichhorn

Modeling and Reasoning with Probabilistics

An Overview of Algorithmic Approaches to Compute Optimum Entropy
Distributions in the Expert System Shell MECore ....................... 61
  Nico Potyka, David Marenke, Engelbert Mittermeier
Ampliative Inference Under Varied Entropy Levels ...................... 75
  Paul Thorn, Gerhard Schurz
Quantitative Measures for Adaptive Object-Oriented World Modeling .... 89
  Achim Kuwertz, Jürgen Beyerer
Abstract. Many have the intuition that for a conditional to be true, there must exist some kind of connection between its antecedent and its consequent. We investigated experimentally the prospects for spelling out the connection in terms of inference. Participants were shown or given by description a series of fourteen numbered color patches ranging from blue to green. They were then asked to evaluate conditionals pertaining to these patches which, in the context of the experiment, are naturally thought of as embodying inferential connections (such as ‘If patch number 5 is blue, then so is patch number 4’). The data we obtained in this experiment are shown to be better explained by a semantics for conditionals that makes inferential connectedness a truth condition than by any of the currently more popular semantics for conditionals. (Based on joint work with Shira Elqayam, Janneke Huitink, David Over, and Henrik Singmann.)
Dynamic Preference Aggregation under Preference Changes

Matthias Thimm
Institute for Web Science and Technologies
Universität Koblenz-Landau, Germany

Abstract. We consider the issue of update in settings for preference aggregation under preference changes. While the traditional problem formulation of preference aggregation assumes a fixed set of preference orders and a fixed set of domain elements we investigate how an aggregated preference order has to be updated when the input orders are dynamic. Our analysis shows that for even for some simple aggregation rules, i.e. for the plurality and the Borda rule, the dynamic setting can be handled more efficiently than recomputing the aggregated preference order from scratch when changes have to be made.

1 Introduction

Preference aggregation [1] deals with the problem of combining the preferences of multiple agents into a joint preference order that reflects the agents’ preferences in a “fair” manner. This field is strongly related to voting systems and social choice theory [2] and has its main application in recommendation and decision making in groups [3, 4]. The traditional setting for preference aggregation is a static one. Given fixed preference orders on a fixed domain one asks for a preference order which jointly represents the former ones. However, in many scenarios both the input preferences and/or the domain under consideration changes. Our motivation for investigating the issue of dynamic preference aggregation stems from recommender systems for social media applications. Consider the situation of visiting a film festival with a mobile recommendation system. In this scenario, one may have preferences regarding movie genres and actors that lead to recommendations to which theater to go next. Furthermore, one may have information on the location of friends and geo-temporal information on the screening of movies which also lead to preference orders, e.g. one would prefer to go to a screening which starts in a couple of minutes or where a lot of friends are present. All these preference orders change over time and so does the aggregated preference order.

In this paper we investigate the issue of preference aggregation under changes of the input preference orders. For that we develop a framework for preference change that distinguishes between two atomic types of changes to a preference order, namely, weakening and strengthening of a specific domain element. Given
a set of preference orders and an aggregation rule we investigate how the ag-
gregated preference order changes under atomic changes of the input preference
orders. We introduce the notion of a dynamic preference aggregator that dy-
namically adapts the aggregated preference order to changes and develop first
dynamic approaches that implement the plurality and the Borda rule for prefer-
ence aggregation. Our analysis of these approaches show that they outperform
a naive re-computation of the aggregated preference order.

The issue of dynamic preference aggregation is closely related to the issue of
bribery [5]. The bribery problem deals with the question of how to (minimally)
adjust voters’ preferences in order to establish some desired aggregated prefer-
ence order. Dynamic preference aggregation is basically the reverse problem as
we investigate how minimal changes of the voters’ preference change the aggre-
gated order. However, to the best of our knowledge the computational questions
arising in reversing the bribery question have not been addressed before explicit-
ly.

The remainder of this paper is structured as follows. In Section 2 we recall
preliminaries on preference representation and preference aggregation. In Sec-
tion 3 we develop our general framework of dynamic preference aggregation un-
der preference change and investigate its general properties. Afterwards we have
a look at specific aggregation rules and their implementation in the dynamic
setting in Section 4. We review some related work in Section 5 and conclude
with a summary and discussion in Section 6.

2 Preferences and Preference Aggregation

Let $O = \{o_1, \ldots, o_n\}$ be a set of outcomes. In order to have a more general
setting, in contrast to the majority of literature on preference aggregation we will
use total preorders, instead of linear orders, to represent preferences. Therefore,
a preference order $\preceq$ on $O$ is a total preorder on $O$, i.e. a relation $\preceq \subseteq O \times O$
that satisfies

1. if $o_1 \succeq o_2$ and $o_2 \succeq o_3$ then $o_1 \succeq o_3$ (transitivity) and
2. for all $o_1, o_2 \in O$ is holds $o_1 \succeq o_2$ or $o_2 \succeq o_1$ (totality)

If $o \preceq o'$ then we say that $o'$ is at least as preferred as $o$. We abbreviate $o \sim o'$ if
both $o \preceq o'$ and $o' \preceq o$, and we abbreviate $o \prec o'$ if $o \preceq o'$ and $o' \not\preceq o$. Let $P_O$ be
the set of all preference orders on $O$. Preference orders can be used to represent
a single individual’s preferences on the possible outcomes (of some action) in $O$.

Example 1. Let $O = \{\text{rock}, \text{pop}, \text{country}, \text{electronic}\}$ be a set of outcomes that
describe the choices for the music genre being played at some event. A possible
preference order $\preceq_{\text{music}}$ on $O$ can be given via

\[
\begin{align*}
\text{country} & \preceq_{\text{music}} \text{pop} \\
\text{country} & \preceq_{\text{music}} \text{electronic} \\
\text{pop} & \preceq_{\text{music}} \text{rock} \\
\text{electronic} & \preceq_{\text{music}} \text{rock}
\end{align*}
\]
In $\preceq_{\text{music}}$ the outcome rock is the most preferred option, both pop and electronic are equally preferred after rock, and country is the least preferred option.

A preference order $\preceq$ on $\mathcal{O}$ can also be concisely characterized by its ranking function $\text{rank}_{\preceq} : \mathcal{O} \rightarrow \mathbb{N}$ defined via

$$\text{rank}_{\preceq}(o) = |\{o' | o' \prec o\}|$$

Note that we have $\text{rank}_{\preceq}(o) \leq \text{rank}_{\preceq}(o')$ if and only if $o \preceq o'$. In the following, we use both the preference order $\preceq$ and its ranking function $\text{rank}_{\preceq}$ interchangeably. For $\preceq_{\text{music}}$ of Example 1 we have

$$\text{rank}_{\preceq_{\text{music}}}(\text{country}) = 0$$
$$\text{rank}_{\preceq_{\text{music}}}(\text{pop}) = 1$$
$$\text{rank}_{\preceq_{\text{music}}}(\text{electronic}) = 1$$
$$\text{rank}_{\preceq_{\text{music}}}(\text{rock}) = 3$$

We define the sets of direct predecessors $\text{pre}_{\preceq}(o)$ and direct successors $\text{suc}_{\preceq}(o)$ of an outcome $o$ as follows:

$$\text{pre}_{\preceq}(o) = \{o' \in \mathcal{O} | o' \prec o \land \neg \exists o'': o' \prec o'' \prec o\}$$
$$\text{suc}_{\preceq}(o) = \{o' \in \mathcal{O} | o \prec o' \land \neg \exists o'': o \prec o'' \prec o'\}$$

We further define the set of most preferred outcomes $\text{top}(\preceq)$ and the set of least preferred outcomes $\text{bot}(\preceq)$ via

$$\text{top}(\preceq) = \{o \in \mathcal{O} | \text{suc}_{\preceq}(o) = \emptyset\}$$
$$\text{bot}(\preceq) = \{o \in \mathcal{O} | \text{pre}_{\preceq}(o) = \emptyset\}$$

In Example 1 we have e.g.

$$\text{suc}_{\preceq_{\text{music}}}(\text{country}) = \{\text{pop, electronic}\}$$
$$\text{top}_{\preceq_{\text{music}}}(\text{country}) = \{\text{rock}\}$$

The field of preference aggregation [1] is concerned with how potentially conflicting preferences of different individuals are to be aggregated in order to come up with a single preference order that reflects the group’s joint preference in a “fair” sense. Formally, a preference aggregator is defined as follows.

**Definition 1.** A preference aggregator $\Theta$ (of order $m \in \mathbb{N}^+$) is a function $\Theta : \mathcal{P}^m \rightarrow \mathcal{P}$.

A vector of preference orders $\preceq \in \mathcal{P}^n$ is also called a preference profile.

In the past 60 years a series of different implementations for preference aggregators have been proposed. The goal of this research is to define a preference aggregator that is, in some sense, “fair”, i.e. reflects the individual preferences in an unbiased and intuitively correct way. Some properties that formalize the notion of “fairness” are as follows.
Pareto-efficiency (PE) For every $o, o' \in O$, if $o \preceq_i o'$ for all $i = 1, \ldots, m$ then $o \preceq o'$ (for preference orders $\preceq_1, \ldots, \preceq_m$ and $\preceq = \Theta(\preceq_1, \ldots, \preceq_m)$).

Non-dictatorship (ND) There is no $i \in \{1, \ldots, m\}$ such that $\preceq_i = \Theta(\preceq_1, \ldots, \preceq_m)$ for every $(\preceq_1, \ldots, \preceq_m) \in \mathcal{P}_O^m$.

Independence of irrelevant alternatives (IIA) If for two profiles $(\preceq_1, \ldots, \preceq_m)$ and $(\preceq'_1, \ldots, \preceq'_m)$ and every $i = 1, \ldots, m$ it holds $o \preceq_i o'$ whenever $o \preceq'_i o'$ then $o \preceq o'$ whenever $o \preceq' o'$ (with $\preceq = \Theta(\preceq_1, \ldots, \preceq_m)$ and $\preceq' = \Theta(\preceq'_1, \ldots, \preceq'_m)$).

Monotonicity (Mon) If for two profiles $(\preceq_1, \ldots, \preceq_m)$ and $(\preceq'_1, \ldots, \preceq'_m)$ we have that $o \preceq_i o'$ implies $o \preceq'_i o'$ then $o \preceq o'$ implies $o \preceq o'$ (with $\preceq = \Theta(\preceq_1, \ldots, \preceq_m)$ and $\preceq' = \Theta(\preceq'_1, \ldots, \preceq'_m)$).

Pareto-efficiency says that if all preference orders agree that $o'$ is at least as preferred as $o$ then $o'$ should also be at least as preferred as $o$ in the joint preference order. The property non-dictatorship says that there is no single individual (the dictator) whose preference order is always adopted as the joint preference order, regardless of what the preference orders actually look like. The property independence of irrelevant alternatives can be best explained in the context of voting. There, this property states that given two elections $A$ and $B$ with voters $1, \ldots, m$, if every voter $i$ prefers candidate 1 to candidate 2 in $A$ whenever he prefers candidate 1 to candidate 2 in $B$, then candidate 1 is preferred to candidate 2 in the outcome of the election $A$ if and only if candidate 1 is preferred to candidate 2 in the outcome of the election $B$. This means that preferences regarding other combinations of candidates are irrelevant for the decision regarding just candidates 1 and 2. Monotonicity basically demands that strengthening the position of an outcome $o$ in one preference order cannot result in a worse situation for $o$ in the aggregated preference order. Arrow’s famous impossibility result [6] states that there is no preference aggregator $\Theta$ which satisfies (PE), (ND), and (IIA).\footnote{Note that the original impossibility result refers to total orders instead of total preorders but a similar result holds for total preorders as well.}

Research in preference aggregation did come up with several preference aggregators that are meaningful under certain circumstances. In this paper, we consider two of the most simple preference aggregators as suitable examples, namely plurality preference aggregation and Borda preference aggregation which are both special cases of scoring rules, cf. [1]. Note that the original definitions for these preference aggregators are given for total orders instead of total preorders as we use here. We adapt those definitions appropriately for total preorders as follows, cf. [7] for some general discussions regarding partially ordered preferences. Let $\preceq = (\preceq_1, \ldots, \preceq_m)$ be a preference profile. The plurality preference aggregator $\Theta_p : \mathcal{P}_O^m \rightarrow \mathcal{P}_O$ and the Borda preference aggregator $\Theta_b : \mathcal{P}_O^m \rightarrow \mathcal{P}_O$.\footnote{Note that the original impossibility result refers to total orders instead of total preorders but a similar result holds for total preorders as well.}
are defined via \( \Theta_p(\succeq) = \preceq_p \) and \( \Theta_b(\succeq) = \preceq_b \), respectively, and
\[
o \preceq_p o' \text{ iff } |\{i \mid o \in \text{top}(\preceq_i)\}| \leq |\{i \mid o' \in \text{top}(\preceq_i)\}|
o \preceq_b o' \text{ iff } \sum_{i=1}^{m} (\text{rank}_{\preceq_i}(o)) \leq \sum_{i=1}^{m} (\text{rank}_{\preceq_i}(o')).
\]

Plurality preference aggregation generalizes majority voting and says, that \( o' \) is at least as preferred as \( o \) if and only if \( o' \) appears at least as often as top element in the preference orders \( \preceq_1, \ldots, \preceq_m \) as \( o \). The intuition behind Borda preference aggregation is that \( o' \) is at least as preferred as \( o \), if the sum of all ranks over all considered preference orders for \( o' \) is smaller or equal to the sum for \( o \). Note, that we generalized the standard definition of the Borda rule by assigning equal rank to equally classified outcomes.

3 Dynamic Preference Aggregation

In the traditional setting for preference aggregation, preference orders are static entities that reflect the preferences of some individual at some given point in time. Furthermore, a preference aggregator is a simple function that takes this static view and returns a coherent joint preference order for the very same point in time. However, in reality preferences are rarely static but change frequently. In the following, we consider the problem of how the joint preference order changes given that one of the input preference orders changes.

In order to describe “change” of preference orders we need some further notation. We focus on specific atomic change operations for preference orders where we only change the preference of a single outcome. More precisely, for a preference order \( \preceq \), an outcome \( o \in \mathcal{O} \), the strengthening of \( \preceq \) by \( o \), denoted by \( \preceq' = \preceq + o \) characterized via

\[
\text{rank}_{\preceq'}(\alpha) = \begin{cases} 
\text{rank}_{\preceq}(\alpha) + |\{o' \mid o' \sim \alpha\} \setminus \{\alpha\}| & \text{(if } \alpha = o) \\
\text{rank}_{\preceq}(\alpha) - 1 & \text{(if } \alpha \in \text{suc}_{\preceq}(o) \text{ and } |\{o' \mid o' \sim o\}| = 1) \\
\text{rank}_{\preceq}(\alpha) & \text{(otherwise)}
\end{cases}
\]

for every \( o' \in \mathcal{O} \). Similarly, the weakening of \( \preceq \) by \( o \), denoted by \( \preceq - o \), is the preference order \( \preceq' = \preceq - o \) characterized via

\[
\text{rank}_{\preceq'}(\alpha) = \begin{cases} 
\text{rank}_{\preceq}(\alpha) - |\{o' \mid o' \in \text{pre}(\alpha)\}| & \text{(if } \alpha = o \text{ and } |\{o' \mid o' \sim o\}| = 1) \\
\text{rank}_{\preceq}(\alpha) + 1 & \text{(if } \alpha \neq o \text{ and } \alpha \sim o) \\
\text{rank}_{\preceq}(\alpha) & \text{(otherwise)}
\end{cases}
\]

for every \( o' \in \mathcal{O} \).
Example 2. We continue Example 1 and assume the music genre \textit{electronic} has to be weakened in $\preceq_{\text{music}}$. Then we have $\preceq_{\text{music}}'=\preceq_{\text{music}}-\text{electronic}$ with

\[
country \preceq_{\text{music}}' \text{ electronic} \preceq_{\text{music}}' \text{ pop} \preceq_{\text{music}}' \text{ rock}
\]

with

\[
\begin{align*}
\text{rank}_{\preceq_{\text{music}}'}(\text{country}) &= 0 & \text{rank}_{\preceq_{\text{music}}'}(\text{electronic}) &= 1 \\
\text{rank}_{\preceq_{\text{music}}'}(\text{pop}) &= 2 & \text{rank}_{\preceq_{\text{music}}'}(\text{rock}) &= 3
\end{align*}
\]

Furthermore, if we weaken \textit{electronic} even further we obtain $\preceq_{\text{music}}''=\preceq_{\text{music}}'-\text{electronic}$ with

\[
country, \text{ electronic} \preceq_{\text{music}}'' \text{ pop} \preceq_{\text{music}}'' \text{ rock}
\]

and

\[
\begin{align*}
\text{rank}_{\preceq_{\text{music}}''}(\text{country}) &= 0 & \text{rank}_{\preceq_{\text{music}}''}(\text{electronic}) &= 0 \\
\text{rank}_{\preceq_{\text{music}}''}(\text{pop}) &= 2 & \text{rank}_{\preceq_{\text{music}}''}(\text{rock}) &= 3
\end{align*}
\]

Strengthening and weakening of outcomes can be regarded as the most basic change operation for a preference order. Some simple observations on strengthening and weakening are summarized in the following proposition. Proofs of technical results are omitted due to space restrictions.

**Proposition 1.** Let $\succeq \in \mathcal{P}_O$ be a preference order.

1. $(\succeq - o) + o = \succeq$ for all $o \notin \text{bot}(\succeq)$,
2. $(\succeq + o) - o = \succeq$ for all $o \notin \text{top}(\succeq)$,
3. for every other preference order $\succeq' \in \mathcal{P}_O$ there are sequences $\langle \pm_1, \ldots, \pm_l \rangle$, $\langle o_1, \ldots, o_l \rangle$ with $\pm_i \in \{+, -\}$, $o_i \in O$ for $i = 1, \ldots, l$ for some $m \in \mathbb{N}$ such that $\succeq = \succeq' \pm_1 o_1 \ldots \pm_l o_l$.
4. computing $\succeq \pm o$ has time complexity $O(n)$.

Items 1.) and 2.) state that strengthening and weakening are inverse operations as long as the element is not a most or least preferred outcome, respectively. Item 3.) states that strengthening and weakening operations are generating operations of the set of preference orders, i.e. every preference order can be represented as applying a series of strengthening/weakening operations to every other preference relation. Finally, item 4.) states that determining $\succeq \pm o$ from $\succeq$ has time complexity $O(n)$.

If $\succeq = \langle \succeq_1, \ldots, \succeq_m \rangle \in \mathcal{P}_O^m$ is a preference profile we abbreviate

$\succeq \pm_i o = \langle \succeq_1, \ldots, \succeq_i \pm o, \ldots, \succeq_m \rangle$

with $\pm \in \{+, -\}$, $o \in O$, and $i = 1, \ldots, m$. The single-step operators + and − can be extended to multi-step operators $+^h$ and $-^h$ for $h \in \mathbb{N}$ via

\[
\begin{align*}
\succeq +^h o &= \succeq + \underbrace{o + \ldots + o}_{h \text{ times}} \\
\succeq -^h o &= \succeq - \underbrace{o - \ldots - o}_{h \text{ times}}
\end{align*}
\]
Some observations relating change operations and preference aggregation are as
follows.

Proposition 2. Let $\preceq$ be a preference profile, $i \in \{1, \ldots, m\}$, $\pm \in \{+,-\}$, $o \in \mathcal{O}$, $\Theta$ a preference aggregator, $\preceq = \Theta(\preceq)$, and $\preceq' = \Theta(\preceq \pm o)$.

1. If $\Theta$ satisfies (IIA) then $o_1 \preceq o_2$ iff $o_1 \preceq' o_2$ for all $o_1, o_2 \in \mathcal{O} \setminus \{o\}$.
2. If $\pm = +$ and $\Theta$ satisfies (Mon) then $o_1 \preceq o_2$ implies $o_1 \preceq' o_2$ for all $o_1 \in \mathcal{O} \setminus \{o\}, o_2 \in \mathcal{O}$.
3. If $\pm = -$ and $\Theta$ satisfies (Mon) then $o_1 \preceq o_2$ implies $o_1 \preceq' o_2$ for all $o_1 \in \mathcal{O}, o_2 \in \mathcal{O} \setminus \{o\}$.

The above proposition states that there are many cases in which the aggregated preference order only changes minimally on atomic changes of an input preference order. For that, it seems beneficial to investigate the computational issue of dynamic preference aggregation. The core concept of our formalism is the dynamic preference aggregator which can naively be defined as follows.

Definition 2. A dynamic preference aggregator $\Lambda$ is a function $\Lambda : \mathcal{P}_\mathcal{O} \times \mathbb{N} \times \{-, +\} \times \mathcal{O} \rightarrow \mathcal{P}_\mathcal{O}$.

Let $\preceq \in \mathcal{P}_\mathcal{O}$ be some preference order that is the result of applying some aggregation rule on the profile $\preceq = (\preceq_1, \ldots, \preceq_m)$. Furthermore, let $i \in \{1, \ldots, m\}$, $\pm \in \{+,-\}$, and $o \in \mathcal{O}$. Then the idea behind a dynamic preference aggregator $\Lambda$ is that $\Lambda(\preceq \pm_i o) = \preceq'$ is the result of first updating the preference profile $\preceq$ via $\preceq \pm_i o$ and then aggregating again to $\preceq'$. However, this update operation should be performed on $\preceq$ directly. Formally, the intuition of dynamic preference aggregation can be phrased as follows.

Definition 3. Let $\Theta$ be a preference aggregator and $\Lambda$ be a dynamic preference aggregator. We say that $\Lambda$ is a faithful representation of $\Theta$ if and only if for all $\preceq = (\preceq_1, \ldots, \preceq_n) \in \mathcal{P}_\mathcal{O}^n$, all $i = 1, \ldots, n$, $\pm \in \{+,-\}$, and all $o \in \mathcal{O}$ it holds

$$\Theta(\preceq \pm i o) = \Lambda(\Theta(\preceq), i, \pm, o)$$

In other words, a faithful representation of a preference aggregator is a dynamic preference aggregator that exhibits the same behavior as the former but adapts to changes of the input preference orders. Given a preference profile $\preceq$ we expect updating $\Theta(\preceq)$ using $\Lambda$ to be same as first updating $\preceq$ and then applying $\Theta$ again. Figure 1 illustrates this relationship between a preference aggregator $\Theta$ and a faithful representation $\Lambda$ using a commuting diagram. However, the definition of a faithful representation is very restricting as it requires a functional dependency between the old result of aggregation and the new one. In general, this demand is not satisfiable for most interesting aggregation operators.

Example 3. Let $\preceq_1, \preceq_2, \preceq_3$ be preference orders on $\mathcal{O} = \{o_1, o_2, o_3\}$ defined via

$$o_3 \preceq_1 o_1 \preceq_1 o_2$$
$$o_2 \preceq_2 o_3 \preceq_2 o_1$$
$$o_3 \preceq_3 o_2 \preceq_3 o_1$$
By using plurality aggregation we obtain $\Theta_p(\preceq_1, \preceq_2) = \Theta_p(\preceq_1, \preceq_3) = \preceq$ with

$$o_1 \sim o_2, o_3 \prec o_1, o_3 \prec o_2.$$ 

That is, we obtain that profiles $\langle \preceq_1, \preceq_2 \rangle$ and $\langle \preceq_1, \preceq_3 \rangle$ are aggregated yielding the same result. Let $\preceq = (\preceq_1, \preceq_2)$, $\preceq' = (\preceq_1, \preceq_3)$ and consider $\preceq = \Theta_p(\preceq + 2o_2)$ resp. $\preceq' = \Theta_p(\preceq' + 2o_2)$ with

$$o_1 \sim o_2, o_3 \prec o_1, o_3 \prec o_2,$$

$$o_1 \preceq o_2, o_3 \prec o_2, o_3 \preceq o_1.$$ 

In summary, we get $\Theta_p(\preceq) = \Theta_p(\preceq')$ but $\Theta_p(\preceq + 2o_2) \neq \Theta_p(\preceq' + 2o_2)$. Therefore, there can be no dynamic preference aggregator $\Lambda_p$ that is a faithful representation of $\Theta_p$.

The above example (unsurprisingly) shows that there is no direct functional dependency between $\preceq$ and $\preceq'$. In order to get to actual approaches for addressing the dynamic preference problem we need to have some more information that is carried from one update to the other.

**Definition 4.** A state-based dynamic preference aggregator $\Delta$ is a pair $\Delta = \langle \iota, \Lambda \rangle$ such that

1. $\iota$ is function $\iota : \mathcal{P}_m^\mathcal{O} \rightarrow S \times \mathcal{P}_\mathcal{O}$ and
2. $\Lambda$ is a function $\Lambda : S \times \mathbb{N} \times \{+, -\} \times \mathcal{O} \rightarrow S \times \mathcal{P}_\mathcal{O}$

where $S$ is some set of states.

The intuition behind a state-based dynamic preference aggregator $\Delta = \langle \iota, \Lambda \rangle$ is as follows. Given a preference profile $\preceq = (\preceq_i_1, \ldots, \preceq_m)$ the function $\iota$ is the *initialization function* that delivers $\iota(\preceq) = (S, \preceq)$ where $\preceq$ is the aggregated preference order of $\preceq$ and $S$ some state (defined in some way suitable for the preference aggregator) that is carried over to the next update. Given a state $S$, $i \in \{1, \ldots, m\}$, $\pm \in \{+, -\}$, and $o \in \mathcal{O}$ the value $\Lambda(S, i, \pm, o) = (S', \preceq')$ then
updates the state \( S \) to \( S' \) and the aggregated order \( \preceq \) to \( \preceq' \), given the change operation \( \pm_i \) on \( o \). Formally, this intuition extends our notion of faithfulness as follows.

**Definition 5.** Let \( \Theta \) be a preference aggregator and \( \Delta = \{i, \Lambda\} \) be a state-based dynamic preference aggregator. We say that \( \Delta \) is a state-based faithful representation of \( \Theta \) if and only if for all \( \preceq = \{\preceq_1, \ldots, \preceq_m\} \in \mathcal{P}_O \), all sequences \( \{\{\pm_1, i_1, o_1\}, \ldots, \{\pm_k, i_k, o_k\}\} \in \{\{\pm\} \times \mathbb{N} \times \mathcal{O}\}^k \) there are states \( S_1, \ldots, S_{k-1} \in S \) such that

\[
(S_1, \Theta(\preceq)) = \iota(\preceq) \\
(S_j, \Theta(\preceq(\pm_1)_{i_1} \cdots (\pm_j)_{i_j} o_j)) = \Lambda(S_{j-1}, i_j, \pm_j, o_j)
\]

for \( j = 2, \ldots, k \).

In the next section we investigate actual approaches for state-based dynamic preference aggregators that effectively implement the preference aggregators discussed before.

## 4 Approaches and Analysis

We first discuss a naive implementation which just applies the original preference aggregator on every change operation.

**Definition 6.** Let \( \Theta \) be a preference aggregator and let \( S_{\text{can}} = \mathcal{P}_O^m \). The canonical state-based dynamic preference aggregator \( \Delta_{\text{can}}^\Theta \) for \( \Theta \) is the pair \( \Delta_{\text{can}}^\Theta = \{\iota_{\text{can}}^\Theta, \Lambda_{\text{can}}^\Theta\} \) with \( \iota_{\text{can}}^\Theta : \mathcal{P}_O^m \rightarrow S_{\text{can}} \times \mathcal{P}_O \) and \( \Lambda_{\text{can}}^\Theta : S_{\text{can}} \times \mathbb{N} \times \{\pm\} \times \mathcal{O} \rightarrow S_{\text{can}} \times \mathcal{P}_O \) defined via

\[
\iota_{\text{can}}^\Theta(\preceq) = (\preceq, \Theta(\preceq)) \\
\Lambda_{\text{can}}^\Theta(\preceq, i, \pm, o) = (\preceq \pm_i o, \Theta(\preceq \pm_i o))
\]

The canonical state-based dynamic preference aggregator simply carries over the whole preference profile from one iteration to the next and applies the preference aggregator in a direct way. This is the baseline approach for solving the dynamic preference aggregation problem but, obviously, the effort required at each iteration should not be necessary. The following proposition follows by construction of the canonical state-based dynamic preference aggregator.

**Proposition 3.** Let \( \Theta \) be a preference aggregator. Then \( \Delta_{\text{can}}^\Theta \) is a state-based faithful representation of \( \Theta \).

**Example 4.** We continue Example 3 and define \( \preceq = (\preceq_1, \preceq_2, \preceq_3) \). For the plurality preference aggregator \( \Theta_p \) we obtain

\[
\iota_{\text{can}}^{\Theta_p}(\preceq) = (\preceq, \preceq)
\]
where $\preceq$ is defined via $o_3 \prec o_2 \prec o_1$. Therefore, $\preceq$ is the initial result of aggregating $\preceq$. Assume now that in the preference order $\preceq_3$ the outcome $o_2$ is to be strengthened, i.e., we have $\preceq_3' = \preceq_3 + o_2$ with $o_3 \preceq_3' o_1, o_2$. Then we have
\[
A_{p}^{\preceq_n}(\preceq, 3, +, o_2) = (\langle \preceq_1, \preceq_2, \preceq_3' \rangle, \preceq')
\]
where $\preceq' = \Theta_p(\preceq_1, \preceq_2, \preceq_3')$ is defined via $o_3 \prec' o_1, o_2$. In this example, a (simple) strengthening of $o_2$ in one of the input preference orders caused a strengthening of $o_2$ in the aggregated order as well.

**Proposition 4.** Let $\Theta$ be a preference aggregator of time complexity $O(f(n, m))$. Then $\iota_{p}^{\preceq_n}$ has time complexity $O(f(n, m))$ and $\Lambda_{p}^{\preceq_n}$ has time complexity $O(n + f(n, m))$. The space complexity for storing a state in $S_{can}$ is $O(nm)$.

The question we are addressing in the following is whether the time complexity $O(n + f(n, m))$ of updating the aggregated preference order can be improved. Proposition 2 already suggested that there are many cases under which the aggregated preference order only changes slightly. We now have a look at some concrete preference aggregators and, first, give a straightforward approach for the plurality aggregator.

**Definition 7.** Let $\preceq = (\preceq_1, \ldots, \preceq_m)$ and let $S_p = P_m \times \mathbb{N}^O$. Define
\[
\iota_p(\preceq) = (\langle \preceq, g \rangle, \Theta_p(\preceq))
\]
with $g : O \to \mathbb{N}$ and $g(o) = |\{i \mid o \in \text{top}(\preceq_i)\}|$. Define
\[
A_p(\langle \preceq, g \rangle, i, \pm, o) = (\langle \preceq \pm_i o, g' \rangle, \preceq')
\]
with $g' : O \to \mathbb{N}$ defined via
\[
g'(o') = g(o') - [o' \in \text{top}(\preceq_i) \land o \notin \text{top}(\preceq_i \pm o)]
+ [o' \notin \text{top}(\preceq_i) \land o \in \text{top}(\preceq_i \pm o)]
\]
and
\[
o_1 \preceq' o_2 \iff g'(o_1) \leq g'(o_2)
\]
Then $\Delta_p = (\iota_p, A_p)$ is called the dynamic plurality preference aggregator.

The above definition avoids re-computing the whole aggregated preference order and only considers changes in the top elements of the preference order that is being changed. More precisely, the value $g(o')$—which stores the number of times each element appears as a top element in the preference orders—is updated by subtracting 1 if the element has been top-ranked in $\preceq_i$ and is no longer top-ranked in $\preceq_i \pm o$ or by adding 1 if the element has not been top-ranked in $\preceq_i$ but is top-ranked in $\preceq_i \pm o$.

\[\text{[P]}\] is the Iverson bracket defined via $[P] = 1$ if $P$ is true and $[P] = 0$ otherwise.

---

11
Proposition 5. $\Delta_p$ is a state-based faithful representation of $\Theta_p$.

Proposition 6. $\iota_p$ has time complexity $O(nm)$ and $\Lambda_p$ has time complexity $O(n)$. The space complexity for storing a state in $S_p$ is $O(nm)$.

Note that the time complexity of computing $\Theta_p(\succeq)$ is $O(nm)$. Thus $\Lambda_p$ has better time complexity ($O(n)$) than the canonical solution $\Lambda_{\Theta_p}^{\text{can}}$ with $O(nm)$, cf. Proposition 4.

Example 5. We continue Example 4. For $\preceq = (\preceq_1, \preceq_2, \preceq_3)$ we obtain

\[ \iota_p(\preceq) = \langle \langle \preceq, g \rangle, \preceq \rangle \]

with $\preceq$ as in Example 4 and $g : O \to \mathbb{N}$ given via $g(o_1) = 2$, $g(o_2) = 1$, and $g(o_3) = 0$. Given that $o_2$ is to be strengthened in $\preceq_3$, i.e. $\preceq'_3 = \preceq_3 + o_2$, we obtain

\[ \Lambda_p((\preceq, g), 3, +, o_2) = \langle \langle \preceq_1, \preceq_2, \preceq'_3 \rangle, g'_1 \rangle, \preceq' \rangle \]

with

\[
\begin{align*}
g'(o_1) &= g(o_1) - [o_1 \in \text{top}(\preceq_3) \land o_1 \notin \text{top}(\preceq'_3)] \\
&\quad + [o_1 \notin \text{top}(\preceq_3) \land o_1 \in \text{top}(\preceq'_3)] = 2 - 0 + 0 = 2 \\
g'(o_2) &= 1 - 0 + 1 = 2 \\
g'(o_3) &= 0 - 0 + 0 = 0
\end{align*}
\]

and therefore $o_3 \prec' o_1, o_2$.

We now turn to the Borda preference aggregator.

Definition 8. Let $\preceq = (\preceq_1, \ldots, \preceq_m)$ and let $S_b = P_O^m \times \mathbb{N}^O$. Define

\[ \iota_b(\preceq) = \langle \langle \preceq, g \rangle, \Theta_b(\preceq) \rangle \]

with $g : O \to \mathbb{N}$ and $g(o) = \sum_{i=1}^m \text{rank}_{\preceq_i}(o)$. Define

\[ \Lambda_b((\preceq, g), i, \pm, o) = \langle \langle \preceq \pm_i o, g' \rangle, \preceq' \rangle \]

with $g' : O \to \mathbb{N}$ defined via

\[ g'(o') = g(o') - \text{rank}_{\preceq_i}(o') + \text{rank}_{\preceq_i \pm o}(o') \]

and

\[ o_1 \preceq' o_2 \quad \text{iff} \quad g'(o_1) \leq g'(o_2) \]

Then $\Delta_b = (\iota_b, \Lambda_b)$ is called the dynamic Borda preference aggregator.

The dynamic Borda preference aggregator is defined in a similar way as the dynamic plurality preference aggregator (in fact, both definitions can easily be generalized to obtain a scheme for dynamic preference aggregators for preference aggregators based on scoring rules). For each $o' \in O$ only the rank changes of $o'$ in $\preceq_i$ are taken into account when updating the value $g(o')$, which stores the sum of all ranks for all $\preceq_j$, $j = 1, \ldots, m$. 

12
Proposition 7. $\Delta_b$ is a state-based faithful representation of $\Theta_b$.

Proposition 8. $\iota_b$ has time complexity $O(nm)$ and $\Lambda_b$ has time complexity $O(n)$. The space complexity for storing a state in $S_b$ is $O(nm)$.

Note that the time complexity of computing $\Theta_b(\preceq)$ is $O(nm)$. Thus $\Lambda_b$ has better time complexity ($O(n)$) than the canonical solution $\Lambda^{can}_{\Theta_b}$ with $O(nm)$, cf. Proposition 4.

Example 6. We continue Example 4. For $\preceq = (\succeq_1, \succeq_2, \succeq_3)$ we obtain
$$\iota_b(\preceq) = \langle\langle \preceq, g \rangle, \preceq \rangle$$
with $\preceq = \Theta_b(\preceq)$ given via $o_3 \prec o_2 \prec o_1$ and $g : O \rightarrow \mathbb{N}$ given via $g(o_1) = 5$, $g(o_2) = 3$, and $g(o_3) = 1$. Given that $o_2$ is to be strengthened $\preceq_3$, i.e. $\preceq'_3 \succeq \preceq_3$ + $o_2$, we obtain
$$\Lambda_b((\preceq, g), 3, +, o_2) = \langle\langle(\preceq_1, \preceq_2, \preceq'_3), g', \preceq' \rangle$$
with
$$g'(o_1) = g(o_1) - \text{rank}_{\preceq_3}(o_1) + \text{rank}_{\preceq'_3}(o_1)$$
$$= 5 - 2 + 1 = 4$$
$$g'(o_2) = 3 - 1 + 1 = 3$$
$$g'(o_3) = 1 - 0 + 0 = 1$$
and therefore $\preceq' = \preceq$.

The computational approaches discussed so far illustrate that in dynamic settings, preference aggregation can be implemented more effectively. This work is the first step towards investigating the issue of dynamic preference aggregation. Current work deals with investigating more complex preference aggregation mechanisms such as the Dodgson and Kemeny rules, cf. [8, 9].

5 Related Work

Preference reasoning and preference aggregation is a very active area within artificial intelligence research, economics, and other fields. However, there are only few works that deal with dynamics in preference aggregation settings. Related to our work is e.g. [10] that deals with influence of one agent’s preferences to other agents’ preferences. This setting also entails some dynamics as preference relations might be changed through influence. Maudet et al. investigate this setting within the framework of CP-nets [3], a specific approach to reason with preferences. They are mainly interested in computational properties in this framework but do not investigate preference change in our more general setting. The work [11] explicitly considers updates to input preference orders and their influence.
to aggregation. They introduce the property of update monotonicity for (static) preference aggregators as a novel property to assess the quality of an aggregator. An aggregator satisfies update monotonicity if under changes of one input preference order towards the aggregated order, the aggregation does not change. Therefore, [11] do not consider general updates and computational approaches.

Furthermore, there are approaches in belief revision that deal with dynamics of epistemic states given in form of preorders. For example, the work [12] considers iterated belief revision based on enriched preference states. There, a preference state is basically a preference order on possible worlds that is revised upon newly received evidence. The work [13] deals with revising a given preference relation with another (partial) one such that the former is modified in a minimal way to incorporate the latter. Although these works also deal with issues related to temporal evolution of (preference) orders they do not address the evolution of the aggregated orders.

Top-k querying [14] deals with effective approaches to determine the best $k$ answers to a (relational) query, given possibly multiple preference orders. In many approaches, the final aggregated order is constructed by incrementally taking more result items into account. Similar to the setting of dynamic preference aggregation the aggregated order is also dynamically updated. However, top-k querying mechanisms consider fixed preference orders and dynamics is only implicitly present by taking more result items into account. The work [15] considers a similar setting of taking more result items with fixed preferences into account but also uses methods from preference aggregation. However, the issue of dynamics is only briefly discussed and not elaborated.

6  Summary and Conclusion

We discussed the issue of dynamics in general settings of preference aggregation under preference change. We introduced the concept of dynamic preference aggregators and investigated the consequences of atomic changes in the input preference orders. Besides some general results on the relationships between the original aggregated order and the updated aggregated order we also established a framework for computational approaches to dynamic preference aggregation. We developed dynamic preference aggregators for two simple aggregation mechanisms, namely plurality and Borda, and discussed their properties.

As mentioned in the introduction the work developed here is applied in the field of social web recommendation systems and a first prototype of a working application is currently under development. Another application of dynamic preference aggregators can also been seen in preference elicitation [16]. Preference elicitation describes the process of iteratively updating an initially empty preference order in order to determine an agent’s preference order. When preference orders of multiple agents have to be elicited and their orders have to be aggregated methods for dynamic preference aggregation can be utilized.
References

Characteristics of Multiple Viewpoints in Abstract Argumentation

Paul E. Dunne¹, Wolfgang Dvořák², Thomas Linsbichler³, Stefan Woltran³

¹ University of Liverpool, UK
² University of Vienna, Austria
³ Vienna University of Technology, Austria

Abstract. The study of extension-based semantics within the seminal abstract argumentation model of Dung has largely focused on definitional, algorithmic and complexity issues. In contrast, matters relating to comparisons of representational limits, in particular, the extent to which given collections of extensions are expressible within the formalism, have been under-developed. As such, little is known concerning conditions under which a candidate set of subsets of arguments are “realistic” in the sense that they correspond to the extensions of some argumentation framework \( AF \) for a semantics of interest. In this paper we present a formal basis for examining extension-based semantics in terms of the sets of extensions that these may express within a single \( AF \). We provide a number of characterization theorems which guarantee the existence of \( AFs \) whose set of extensions satisfy specific conditions and derive preliminary complexity results for decision problems that require such characterizations.

1 Introduction

The last 15 years have seen an enormous effort to design, compare, and implement different semantics for Dung’s abstract argumentation frameworks [13], \( AFs \) for short. Not at least this extensive study made argumentation a main topic of current AI research [7, 19]. Surprisingly, a systematic comparison of their capability in terms of multiple extensions, and thus their power in modelling multiple viewpoints with a single \( AF \) has been neglected so far. Understanding which extensions can, in principle, go together when a framework is evaluated with respect to a semantics of interest not only clarifies the “strength” of that semantics but also is a crucial issue in several applications.

In this work, we close this gap by studying the signatures

\[
\Sigma_\sigma = \{ \sigma(F) \mid F \text{ is an AF} \},
\]

of several important semantics \( \sigma \) namely naive, preferred, semi-stable, stage, and stable semantics [13, 20, 10]. Finding simple criteria to decide whether a set \( S \) is contained in \( \Sigma_\sigma \) for different semantics \( \sigma \) is essential in many aspects.

First, it adds to the comparison of semantics (see, e.g., [3]) by means of different properties. So far these properties mostly focused on the aspects of a single extension

⋆ This research has been supported by the Austrian Science Fund (FWF) through projects I1102 and P25521.
$S \in \mathcal{S}$ rather than on a set $S$ thereof. An obvious exception is incomparability (the sets in $\mathcal{S}$ are not proper subsets of each other), but as we will see, all of the standard semantics put additional (yet different) requirements on $\mathcal{S}$ in order to be contained in the signature.

Second, our results are important for constructing $\mathcal{AF}s$. Indeed, knowing whether a set $\mathcal{S}$ is contained in $\Sigma_\sigma$ is a necessary condition which should be checked before actually looking for an $\mathcal{AF}F$ which realizes $\mathcal{S}$ via $\sigma$, i.e. $\sigma(F) = \mathcal{S}$. This is of high importance when dynamic aspects of argumentation are considered [18]. As an example, suppose a framework $F$ possesses as its $\sigma$-extensions a set $\mathcal{S}$ and one asks for an adaptation of the framework $F$ such that its $\sigma$-extensions are given by $\mathcal{S} \cup \{E\}$, i.e. one extension shall be added. Before starting to think how the adapted framework should look like, it is obviously crucial to know whether an appropriate framework exists at all, i.e. whether $\mathcal{S} \cup \{E\} \in \Sigma_\sigma$. In a recent paper on revision of $\mathcal{AF}s$ [12], the authors circumvent this issue by allowing revision to result in a set of $\mathcal{AF}s$ such that the union of their extensions provides the desired outcome. Our results here provide exact conditions under which circumstances their approach can be reduced to single $\mathcal{AF}s$ as an outcome of a given revision.

Finally, we note a connection to instantiation-based argumentation [9], where the concept of rationality postulates plays an important role as does the underlying principle of evaluating argumentation semantics in generic terms. Our results on signatures show that, for a given semantics, certain outcomes (i.e. collections of extensions) are impossible to achieve independent of the concrete way the instantiation process is carried out.

Related work includes studies on enforcing [5, 6] certain outcomes, where the task is to modify $\mathcal{AF}s$ in such a way that desired arguments become acceptable. However, the issue of multiple extensions is not covered. In fact, the work which is closest to our investigations are studies of intertranslatability issues [15, 17], where signatures of semantics are put in relation to each other. More precisely, if there is a translation such that $\theta$-extensions of the transformed $\mathcal{AF}$ coincide with the $\sigma$-extensions of the original $\mathcal{AF}$, then $\theta$ is at least as expressive as $\sigma$, that is $\Sigma_\sigma \subseteq \Sigma_\theta$ in our terms. These results, however, do not tell us anything about the actual contents of $\Sigma_\sigma$ and $\Sigma_\theta$.

To summarise, the main contributions of our work are:

- We first identify necessary conditions any set of extensions under a given semantics $\sigma$ satisfies. This not only guides us towards the exact characteristics for the signature of $\sigma$, but also determines those sets of extensions that are impossible to be jointly expressed with one $\mathcal{AF}$.
- Then, we provide sufficient conditions for a set of extensions to be realizable under a given semantics $\sigma$. For any such realizable set $\mathcal{S}$ of extensions, we moreover provide constructions of canonical frameworks which have $\mathcal{S}$ as their $\sigma$-extensions. Together with the already provided necessary conditions, these realizability results yield exact characteristics of the signatures for the considered semantics.
- We also touch upon optimization issues and strengthen the concept of realizability in such a way that we want to find an $\mathcal{AF}F$ which is solely built from arguments occurring in $\mathcal{S}$ and delivers $\sigma(F) = \mathcal{S}$ (hence, no additional arguments to express
\(S\) are required). We show that for naive semantics each \(S \in \Sigma_{naive}\) can be strictly realized, while this is not the case for the other semantics.

- One particular application of our results is the problem of recasting, i.e. to decide whether the \(\sigma\)-extensions of a given \(AF\) can be expressed via a different semantics \(\theta\). We give some preliminary complexity results of the recasting problem, which go up to \(\Pi^P_2\)-completeness.

## 2 Preliminaries

In what follows, we briefly recall the necessary background on abstract argumentation. For an excellent recent overview, we refer to [1].

Throughout the paper we assume a countably infinite domain \(A\) of arguments. An argumentation framework (AF) is a pair \(F = (A, R)\) where \(A \subseteq A\) is a non-empty, finite set of arguments and \(R \subseteq A \times A\) is the attack relation. The collection of all AFs is given as \(AF_A\). We write \(a \rightarrow_R b\) for \((a, b) \in R\) and \(S \rightarrow_R a\) (resp. \(a \rightarrow_R S\)) if \(\exists s \in S\) such that \(s \rightarrow_R a\) (resp. \(a \rightarrow_R s\)). We drop subscript \(R\) in \(\rightarrow\) if there is no ambiguity. For \(S \subseteq A\), the range of \(S\) (wrt. \(R\)), denoted \(S^+\), is the set \(S \cup \{b \mid S \rightarrow_R b\}\).

Given \(F = (A, R)\), an argument \(a \in A\) is defined (in \(F\)) by a set \(S \subseteq A\) if for each \(b \in A\), such that \(b \rightarrow_R a\), also \(S \rightarrow_R b\). A set \(T\) of arguments is defined (in \(F\)) by \(S\) if each \(a \in T\) is defined by \(S\) (in \(F\)). The following result is in spirit of Dung’s fundamental lemma. We will need it later.

**Lemma 1.** Given an AF \(F = (A, R)\) and two sets of arguments \(S, T \subseteq A\). If \(S\) defends itself in \(F\) and \(T\) defends itself in \(F\), then \(S \cup T\) defends itself in \(F\).

Given an AF \(F = (A, R)\), a set \(S \subseteq A\) is conflict-free (in \(F\)), if there are no arguments \(a, b \in S\), such that \((a, b) \in R\). We denote the set of all conflict-free sets in \(F\) as \(cf(F)\). \(S \in cf(F)\) is called admissible (in \(F\)) if \(S\) defends itself. We denote the set of admissible sets in \(F\) as \(adm(F)\).

The semantics we focus in this work are the naive, stable, preferred, stage, and semi-stable extensions. Given \(F = (A, R)\) they are defined as subsets of \(cf(F)\) as follows:

- \(S \in naive(F)\), if there is no \(T \in cf(F)\) with \(T \supset S\)
- \(S \in stb(F)\), if \(S \rightarrow a\) for all \(a \in A \setminus S\)
- \(S \in pref(F)\), if \(S \in adm(F)\) and \(\exists T \in adm(F)\) s.t. \(T \supset S\)
- \(S \in stage(F)\), if \(\exists T \in adm(F)\) with \(T^+ \supset S^+_R\)
- \(S \in sem(F)\), if \(S \in adm(F)\) and \(\exists T \in adm(F)\) s.t. \(T^+_R \supset S^+_R\)

The objects of our interest are the signatures of a semantics.

**Definition 1.** The signature \(\Sigma_\sigma\) of a semantics \(\sigma\) is defined as

\[
\Sigma_\sigma = \{\sigma(F) \mid F \in AF_A\}
\]

For characterizing the signatures of the semantics of our interest we will make frequent use of the following concepts.

**Definition 2.** Given \(S \subseteq 2^A\), \(Args_S\) denotes \(\bigcup_{S \in S} S\) and \(Pairs_S\) denotes \(\{(a, b) \mid \exists S \in S : \{a, b\} \subseteq S\}\). \(S\) is called an extension-set (over \(A\)) if \(Args_S\) is finite.

As is easily observed, for all considered semantics \(\sigma\) each element \(S \in \Sigma_\sigma\) is an extension-set.
3 Properties of Argumentation Semantics

Our ultimate goal is to characterize the signatures of the semantics under consideration. In this section, we provide necessary conditions for an extension-set \( S \) to be in the signature. We do so by abstracting away from the syntactical structure of a given \( AF \); instead we provide characterizations for the sets \( \sigma(F) \). The first properties, which we define next, enable us to characterize conflict-free sets and naive, stable and stage extensions.

**Definition 3.** Let \( S \subseteq 2^A \). The downward-closure, \( dcl(S) \), of \( S \) is given by \( \{ S' \subseteq S \mid S \in S \} \). Further we call \( S \)
- downward-closed if \( S = dcl(S) \);
- tight if for all \( S \in S \) and \( a \in Args_S \) it holds that if \( S \cup \{ a \} \notin S \) then there exists an \( s \in S \) such that \( (a, s) \notin Pairs_S \);
- incomparable if all elements \( S \in S \) are pairwise incomparable, i.e. for each \( S, S' \in S, S \subseteq S' \) implies \( S = S' \).

In words, an extension-set is downward-closed, if for each element of the extension-set, all subsets of this element are in the extension-set too. The notion of being tight, in a way, limits the multitude of incomparable elements of an extension-set.

**Proposition 1.** For each \( AF \) \( F = (A, R) \),
1. \( cf(F) \) is non-empty, downward-closed and tight;
2. \( naive(F) \) is non-empty, incomparable and its downward-closure is tight;
3. \( stage(F) \) is non-empty, incomparable and tight;
4. \( stb(F) \) is incomparable and tight.

**Proof.** The properties of being non-empty and incomparable are clear. Likewise, it is easy to see that \( cf(F) = dcl(cf(F)) \).

To show that \( cf(F) \) is tight let \( S \in cf(F) \) and \( a \in Args_{cf(F)} \), such that \( S \cup \{ a \} \notin cf(F) \). It follows that \( S \neq \emptyset \). Moreover there exists an argument \( s \in S \) such that \( s \mapsto a \) or \( a \mapsto s \). Then \( \{ a, s \} \notin cf(F) \) and since \( cf(F) \) is downward-closed, \( \{ a, s \} \subseteq T \) for any \( T \in cf(F) \). It follows that \( (a, s) \notin Pairs_{cf(F)} \).

Next, observe that \( dcl(naive(F)) = cf(F) \). It follows that \( dcl(naive(F)) \) is tight.

Also note that if a set \( S \subseteq 2^A \) is tight, then the subset-maximal elements in \( S \) form a tight set \( S' \) too (since for each \( S \in S' \) and \( a \inArgs_{S'} \), if \( S \cup \{ a \} \notin S' \) then \( S \cup \{ a \} \notin S \) and moreover, \( Pairs_S = Pairs_{S'} \)). In other words, since \( dcl(naive(F)) \) is tight, it follows that \( naive(F) \) is tight. Finally, for each incomparable \( S \subseteq 2^A \) it holds that if \( S \) is tight then \( S' \) is tight for each \( S' \subseteq S \). Using \( stb(F) \subseteq stage(F) \subseteq naive(F) \), the result thus follows.

\( \square \)

**Example 1.** For the \( AF \) \( F \) in Figure 1, we have \( S = stb(F) = stage(F) = \{ \{ a_1, b_2, b_3 \}, \{ a_2, b_1, b_3 \}, \{ a_3, b_1, b_2 \} \} \). \( S \) is indeed tight. Take, for instance, \( E = \{ a_1, b_2, b_3 \} \); then none of \( (a_1, b_1), (a_1, a_2), (a_1, a_3) \) is contained in \( Pairs_S \). The other two extensions behave in a symmetric way. However, \( dcl(S) \) is not tight. In fact, \( \{ b_2, b_3 \} \in dcl(S) \) and now for \( b_1, \{ b_1, b_2, b_3 \} \notin dcl(S) \), but \( (b_1, b_2) \) and \( (b_1, b_3) \) are contained in \( Pairs_{dcl(S)} = Pairs_S \). By Proposition 1, thus no \( AF \) with \( naive(G) = S \) exists (note that \( naive(F) = S \cup \{ \{ b_1, b_2, b_3 \} \} \)).
We now turn to admissible sets.

**Definition 4.** A set $S \subseteq 2^A$ is called adm-closed if for each $A, B \in S$ the following holds: if $(a, b) \in \text{Pairs}_S$ for each $a, b \in A \cup B$, then also $A \cup B \in S$.

The property adm-closed is related to aforementioned properties as follows:

**Lemma 2.** For any extension-set $S \subseteq 2^A$ it holds that if $S$ is downward-closed and tight, then $S$ is adm-closed.

The reverse of Lemma 2 does not hold, i.e. there is an extension-set (e.g. $\{\{a,b\}, \{a,c,e\}, \{b,d,e\}\}$), which is adm-closed, but not tight. The following proposition gives necessary conditions for sets of extensions obtained from the admissible semantics.

**Proposition 2.** For each AF $F = (A, R)$, $\text{adm}(F)$ is an adm-closed extension-set containing $\emptyset$.

**Proof.** By definition $\emptyset$ is always admissible. We show that $\text{adm}(F)$ is adm-closed. Towards a contradiction, assume $B, C \in \text{adm}(F)$ such that for all $b, c \in B \cup C$, $(b, c) \in \text{Pairs}_{\text{adm}(F)}$, but $B \cup C \notin \text{adm}(F)$. From Lemma 1 we know that $B \cup C$ defends itself in $F$. So for $B \cup C \notin \text{adm}(F)$ there must be a conflict in $B \cup C$, i.e. $\exists (b, c) \in R$ such that $(b, c) \subseteq B \cup C$. But then, for all $D \in \text{adm}(F)$, $\{b, c\} \nsubseteq D$. Hence, $(b, c) \notin \text{Pairs}_{\text{adm}(F)}$, a contradiction. \qed

The next property characterizes preferred and semi-stable semantics.

**Definition 5.** A set $S \subseteq 2^A$ is pref-closed if for each $A, B \in S$, $A \neq B$, there exist $a, b \in A \cup B$ such that $(a, b) \notin \text{Pairs}_S$.

It is easy to verify that each pref-closed extension-set is incomparable. Moreover, for an incomparable set, pref-closed is a stricter notion than tight. Lemma 3 together with Example 2 will show this.

**Lemma 3.** For a set $S \subseteq 2^A$ it holds that if $S$ is incomparable and tight, then $S$ is pref-closed.

**Proof.** Consider some incomparable and tight extension-set $S \subseteq 2^A$ and assume that $S$ is not pref-closed. That means that there are some $A, B \in S$ with $A \neq B$ such that $\forall a, b \in (A \cup B) : (a, b) \in \text{Pairs}_S$. Since $S$ is incomparable, $B \neq \emptyset$ and $\forall b \in B : (A \cup \{b\}) \notin S$. Considering an arbitrary $b \in B$ we get $\exists a \in A : (a, b) \notin \text{Pairs}_S$ by the fact that $S$ is tight, a contradiction to $\forall a, b \in (A \cup B) : (a, b) \in \text{Pairs}_S$. \qed
We relate the notions of adm- and pref-closed and then show our final characterization result.

**Lemma 4.** A set $S \subseteq 2^A$ is pref-closed iff it is incomparable and adm-closed.

**Proof.** Let $S$ be incomparable and adm-closed and $A, B \in S$. If $A \neq B$, then $A \cup B \notin S$ (by incomparability). Since $S$ is adm-closed, there exist $a, b \in A \cup B$ such that $(a, b) \notin \text{Pairs}_S$. It follows that $S$ is pref-closed. Now consider a set $S \subseteq 2^A$ not incomparable, i.e. $\exists A, B \in S : A \subset B$. But then for all $a, b \in A \cup B : (a, b) \in \text{Pairs}_S$ and thus $S$ is not pref-closed. Finally consider an incomparable $S$ which is not adm-closed. Then there are $A, B \in S$ such that for all $a, b \in A \cup B : (a, b) \in \text{Pairs}_S$ and again $S$ is not pref-closed. \hfill $\Box$

**Proposition 3.** For each AF $F = (A, R)$, $\sigma \in \{\text{pref}, \text{sem}\}$, $\sigma(F)$ is a non-empty, pref-closed extension-set.

**Proof.** By definition both semantics $\sigma \in \{\text{pref}, \text{sem}\}$ always propose at least one extension. Since $\text{sem}(F) \subseteq \text{pref}(F)$ holds for all AFs $F$, it is sufficient to show that $\text{pref}(F)$ is pref-closed. Towards a contradiction, let $B, C \in \text{pref}(F) \ (B \neq C)$, such that for all $a, b \in B \cup C$, $(a, b) \in \text{Pairs}_{\text{pref}(F)}$. It follows that $B \cup C \in \text{cf}(F)$ and by Lemma 1, $B \cup C \in \text{adm}(F)$. Since $B \cup C \supset B$, this is a contradiction to $B \in \text{pref}(F)$. \hfill $\Box$

**Example 2.** Consider the AF $F$ in Figure 2 and let $A = \{a, b\}$, $B = \{a, c, e\}$, $C = \{b, d, e\}$, and $S = \{A, B, C\}$. We have $\text{pref}(F) = \text{sem}(F) = S$ and, indeed, $S$ is pref-closed: $b, c \in A \cup B$ and $(b, c) \notin \text{Pairs}_S$; $a, d \in A \cup C$ and $(a, d) \notin \text{Pairs}_S$; $c, d \in B \cup C$ and $(c, d) \notin \text{Pairs}_S$. However, we also observe that $S$ is not tight, since $A \cup \{e\} \notin S$ but both $(a, e)$ and $(b, e)$ are contained in $\text{Pairs}_S$.

## 4 Realizability and Signatures

In the previous section we have given necessary characteristics for the extension-sets $S \in \Sigma_\sigma$, where $\sigma \in \{\text{cf, adm, naive, stb, stage, pref, sem}\}$ are the semantics of our interest. Now we will show that these characteristics are also sufficient. To this end, we require the concept of realizability. In words, an extension-set $S \subseteq 2^A$ is $\sigma$-realizable if there exists an AF $F \in \text{AF}_A$, such that $\sigma(F) = S$. This turns our characterizations into the desired characterizations for $\Sigma_\sigma$.

We start with the following concept of a canonical argumentation framework, which will underlie all subsequent results on realizability.
Definition 6. Given an extension-set \( S \), we define the canonical argumentation framework for \( S \) as
\[
F^\text{cf}_S = (\text{Args}_S, (\text{Args} \times \text{Args}) \setminus \text{Pairs}_S).
\]

The underlying idea for the framework is simple. Whenever two arguments occur jointly in a set \( S \subseteq T \), we must not draw a relation between these two arguments; otherwise we do so. Thus, for any \( S \), \( F^\text{cf}_S \) is symmetric and does not contain any self-attacking arguments. For \( T = \{a_1, b_2, b_3\}, \{a_2, b_1, b_3\}, \{a_3, b_1, b_2\}, \{b_1, b_2, b_3\} \), \( F^\text{cf}_T \) has the same structure as the AF from Figure 1, but with all attacks being symmetric (in fact, orientation does not matter for naive semantics) and naive\((F^\text{cf}_T) = T \). When we consider \( S = \{a_1, b_2, b_3\}, \{a_2, b_1, b_3\}, \{a_3, b_1, b_2\} \), i.e. \( S = T \setminus \{b_1, b_2, b_3\} \), we obtain the same framework \( F^\text{cf}_T = F^\text{cf}_S \). In terms of naive semantics, this is not problematic since \( S \) (as discussed in Example 1) cannot be realized via naive semantics, at all. However, this observation readily suggests that realizing \( S \) with, say, stable semantics, requires additional concepts. We will come back to this issue, but first state some formal results on the canonical framework.

Proposition 4. For each non-empty, downward-closed and tight extension-set \( S \) it holds that \( \text{cf}(F^\text{cf}_S) = S \).

Proof. Let \( F^\text{cf}_S = (\text{Args}_S, R^\text{cf}_S) \). To show \( \text{cf}(F^\text{cf}_S) \subseteq S \), observe that for each \( E \subseteq \text{cf}(F^\text{cf}_S) \), \((a, b) \in \text{Pairs}_S \) for all \( a, b \in E \), by construction of \( R^\text{cf}_S \). Now suppose there exists \( E' \subset \text{cf}(F^\text{cf}_S) \) such that \( E' \notin S \). Wlog. let \( E' \subseteq \text{minimal with this property. Then } E' = S \setminus \{c\} \) for some \( S \subseteq S \). As \( S \) is tight and \( c \in \text{Args}_S \) by construction of \( F^\text{cf}_S \), there is an \( s \in S \) such that \((s, c) \notin \text{Pairs}_S \), a contradiction to the above observation. To show \( \text{cf}(F^\text{cf}_S) \supseteq S \), let \( S \subseteq S \). All \( a, b \in S \) are contained as pairs \((a, b) \in \text{Pairs}_S \), thus by construction, \((a, b) \notin R^\text{cf}_S \). Hence \( S \in \text{cf}(F^\text{cf}_S) \). \( \square \)

We approach the characterization for naive-realizable sets by the following result, which will be useful later.

Lemma 5. For each incomparable and tight extension-set \( S \) it holds that \( S \subseteq \text{naive}(F^\text{cf}_S) \).

Proof. Assume there is an \( S \subseteq S \) such that \( S \notin \text{naive}(F^\text{cf}_S) \). If \( S \notin \text{cf}(F^\text{cf}_S) \), we contradict the above assumption. Thus \( S' \subseteq S \) such that \( S' \notin \text{cf}(F^\text{cf}_S) \). Then by construction of \( F^\text{cf}_S \) \( \forall a, b \in S' : (a, b) \in \text{Pairs}_S \). Since \( S \) is tight also \( S' \subseteq S \), a contradiction to \( S \) being incomparable. \( \square \)

We are now ready to give the full characterization.

Proposition 5. For each incomparable and non-empty extension-set \( S \), where \( \text{dcl}(S) \) is tight it holds that \( \text{naive}(F^\text{cf}_S) = S \).

Proof. Since \( \text{dcl}(S) \) is surely downward-closed, as well as tight and non-empty by definition, we know from Proposition 4 that \( \text{cf}(F^\text{cf}_S) = \text{dcl}(S) \) (note that \( F^\text{dcl}(S) = F^\text{cf}_S \)). By construction of \( \text{dcl}(S) \) the \( \subseteq \)-maximal sets in \( \text{dcl}(S) \) are the sets \( S \subseteq S \) (\( S \) is incomparable by assumption) and as naive sets are just \( \subseteq \)-maximal conflict-free, \( \text{naive}(F^\text{cf}_S) = S \). \( \square \)
So far, in order to realize a set $S$ we used a framework from $AF_A$ of the form $(A, R)$ with $A = \text{ArgS}_S$. For the subsequent results we require, in general, frameworks with $\text{ArgS}_S \subset A$. In the next section, we will show that this cannot be avoided. For the moment, we recall that $A$ is infinite, hence there are always enough arguments available in $A$.

Let us proceed with stable and stage semantics. Stable semantics are the only semantics that can realize $S = \emptyset$. Note that $S = \emptyset$ is easily $stb$-realizable, for instance with the framework $(\{a\}, \{(a, a)\})$. In Proposition 1 the only difference between stable and stage semantics was the case $S = \emptyset$. The next result shows that this indeed is the only difference between the signatures for stable and stage semantics.

The idea of the construction used in the forthcoming proof is to suitably extend the canonical framework from Definition 6 such that undesired stable and stage extensions are excluded. Coming back to our example with $S = \{\{a_1, b_2, b_3\}, \{a_2, b_1, b_3\}, \{a_3, b_1, b_2\}\}$, recall that $F_S^{ef}$ had one such undesired extension, $E = \{b_1, b_2, b_3\}$. To get rid of it we add a new argument which is attacked by all other sets from $S$ but not by $E$, see Figure 3 for illustration.

**Proposition 6.** For each non-empty, incomparable and tight extension-set $S$, there exists an $AF F$ such that $stb(F) = stage(F) = S$.

**Proof.** Since $S$ is non-empty, showing existence of an $AF F$ with $stb(F) = S$ is sufficient (for each $F$ with $stb(F) \neq \emptyset$, $stb(F) = stage(F)$ holds).

By Lemma 5 we already know that $S \subseteq \text{naive}(F_S^{ef})$. Let $\mathcal{X} = \text{naive}(F_S^{ef}) \setminus S$ and consider the $AF$ extending $F_S^{ef}$ as follows: $F_S' = (\text{ArgS}_S \cup \{E \mid E \in \mathcal{X}\}, R_S')$ with $R_S' = \{((\text{ArgS}_S \times \text{ArgS}_S) \setminus \text{Pairs}_S) \cup \{(E, E), (a, E) \mid E \in \mathcal{X}, a \in \text{ArgS}_S \setminus E\}\}$ (this construction is borrowed from [15]). We show that $stb(F_S') = S$.

$\text{stb}(F_S') \subseteq S$: Let $E \in stb(F_S')$. As all new arguments $E$ are self-attacking, also $E \in \text{naive}(F_S') = \text{naive}(F_S^{ef}) = \mathcal{X} \cup S$. If $E \in \mathcal{X}$, by construction of $F_S'$, $E \not\Rightarrow \bar{E}$ and also $E \notin \text{stb}(F_S')$. Hence it must be that $E \in S$.

$\text{stb}(F_S') \supseteq S$: Let $E \in S$. By Lemma 5, $E \in \text{naive}(F_S^{ef})$, and, as $F_S^{ef}$ is symmetric, $E \in stb(F_S^{ef})$. Now consider $F_S'$. As we do not change attacks between the arguments $\text{ArgS}_S$, $E \in \text{naive}(F_S')$ and $E$ attacks all arguments in $\text{ArgS}_S \setminus E$. Now consider an arbitrary argument $E'$ for $E' \in \mathcal{X}$. $E'$ is attacked by all arguments $a \in \text{ArgS}_S \setminus E'$ and

---

4 Recall that in every symmetric $AF F$ it holds that $\text{naive}(F) = stb(F) = stage(F)$. 

---

Fig. 3. Excluding the naive extension $\{b_1, b_2, b_3\}$ from $F_S^{gf}$. 

---
as $E, E'$ are both naive sets (and thus incomparable) at least one of these arguments must be contained in $E$. Hence $E \in \text{stb}(F'_S)$ follows. 

Towards a suitable canonical AF for admissibility-based semantics we introduce the following technical concept.

**Definition 7.** Given an extension-set $S$, the defense-formula $D_a^S$ of an argument $a \in \text{Args}_S$ is $\top$ if $\{a\} \in S$ and

$$\bigvee_{S \in S \text{ s.t. } a \in S} \bigwedge_{s \in S \setminus \{a\}} s$$

otherwise. $D_a^S$ converted to (a logical equivalent) conjunctive normal form in clause-form is then called CNF-defense-formula $C_a^S$ of $a$ (in $S$).

Intuitively, $D_a^S$ describes the conditions for the argument $a$ being in an extension. The variables coincide with the arguments. Each disjunct represents a set of arguments which jointly allows $a$ to “join” an extension, i.e. represents a collection of arguments defending $a$.

**Example 3.** Consider $T = \{\{a\}, \{b, c\}, \{a, c, d\}\}$. Then $D_a^T = \top$, $D_b^T = c$, $D_c^T = b \lor (a \land d)$, and $D_d^T = a \land c$. The corresponding CNF-defense-formulas are given as $C_a^T = \{\}$, $C_b^T = \{\{c\}\}$, $C_c^T = \{\{a, b\}, \{b, d\}\}$, and $C_d^T = \{\{a, c\}\}$.

The following lemma shows that the (CNF-)defense-formula for any argument $a$ really captures the intuition of describing which arguments it takes for $a$ in order to join an element of the given extension-set.

**Lemma 6.** Given an extension-set $S$ and an argument $a \in \text{Args}_S$, for each $S' \subseteq \text{Args}_S$ with $a \in S$ the following holds: $(S \setminus \{a\})$ is a model of $D_a^S$ (resp. $C_a^S$) iff there exists an $S' \subseteq S$ with $a \in S'$ such that $S' \subseteq S$.

**Proof.** The if-direction follows straight by definition of $D_a^S$.

To show the only-if-direction consider some $S \subseteq \text{Args}_S$ with $a \in S$ where $S \setminus \{a\}$ is a model of $D_a^S$. If $D_a^S = \top$ then by Definition 7 it holds that $\{a\} \in S$. For $S \setminus \{a\}$ to be a model of $D_a^S \neq \top$, there must be some disjunct $\delta \in D_a^S$, whose elements form a subset of $S \setminus \{a\}$. Consider such a term $\delta \in D_a^S$. Then by construction of $D_a^S$ there is some $S' \subseteq S$ with $a \in S'$, where $S' \setminus \{a\}$ coincides with the elements of $\delta$. So $S' \subseteq S$.

Since $D_a^S \equiv C_a^S$, these formulas can be used interchangeably in this context.

Having at hand a formula for each argument, where its models coincide with the sets of arguments that defend this original argument, we can give the construction of our canonical defense-argumentation-framework.

**Definition 8.** Given an extension-set $S$, the canonical defense-argumentation-framework $F^\text{def}_S = (A^\text{def}_S, R^\text{def}_S)$ extends the canonical AF $F^\text{def}_S = (\text{Args}_S, R^\text{def}_S)$ as follows:

$$A^\text{def}_S = \text{Args}_S \cup \bigcup_{a \in \text{Args}_S} \{\alpha_{a, \gamma} \mid \gamma \in C_a^S\}, \text{ and}$$

$$R^\text{def}_S = R^\text{def}_S \cup \bigcup_{a \in \text{Args}_S} \{(b, \alpha_{a, \gamma}), (\alpha_{a, \gamma}, \alpha_{a, \gamma}), (\alpha_{a, \gamma}, a) \mid \gamma \in C_a^S, b \in \gamma\}.$$
$F^\text{def}_S$ consists of all arguments given in the extension-set plus a certain amount of additional arguments, $\alpha_{a,c}$. Each $\alpha_{a,c}$ attacks argument $a$ and is attacked by all arguments occurring as literals in clause $c$ of the CNF-defense-formula of $a$. So in $F^\text{def}_S$ for $a$ to be defended from $\alpha_{a,c}$ it takes at least one argument of these occurring as atoms in clause $c$ of $C^S_a$, simulating the intended meaning of the defense-formulas.

The following proposition shows that, given any extension-set $S$, each element of $S$ is admissible in the canonical defense-argumentation-framework $F^\text{def}_S$.

**Proposition 7.** For each extension-set $S$ it holds that $S \subseteq \text{adm}(F^\text{def}_S)$.

**Proof.** Let $S \in S$. If $S = \emptyset$, the assertion trivially holds. If $S = \{a\}$, then $C^S_a$ is the empty set of clauses. By definition of $R^\text{def}_S$, $a$ is then defended in $F^\text{def}_S$ and thus $S \in \text{adm}(F^\text{def}_S)$. Thus let $S \in S$ contain at least two arguments. By construction, $S$ is conflict-free in $F^\text{def}_S$. It remains to show that each $s \in S$ is defended by $S$ in $F^\text{def}_S$.

Proposition 7. For each extension-set $S$ it holds that $S \subseteq \text{adm}(F^\text{def}_S)$.

Let $s \in S$. If $\{s\} \in S$, we know from the one-element case that $\{s\} \in \text{adm}(F^\text{def}_S)$, so $s$ is defended. On the other hand, if $\{s\} \notin S$, we know from Lemma 6 that $S \setminus \{s\}$ is a model of $D^S_s$ as well as of $C^S_s$. Hence, each clause $\gamma \in C^S_s$ contains at least one variable $t_\gamma \in S \setminus \{s\}$. Thus, by construction of $R^\text{def}_S$, $S \setminus \{s\} \mapsto \alpha_{s,\gamma}$ for each $\gamma \in C^S_s$, i.e. $S \setminus \{s\}$ defends $s$ in $F^\text{def}_S$. \hfill \Box

While $S \subseteq \text{adm}(F^\text{def}_S)$ holds for any extension-set, $F^\text{def}_S$ may contain additional admissible sets which do not occur in $S$. In order to ensure that $\text{adm}(F^\text{def}_S)$ coincides with $S$ it takes $S$ to be adm-closed and to contain $\emptyset$. Before showing this we give an example.

**Example 4.** Again consider the extension-set $T = \{\{a\}, \{b, c\}, \{a, c, d\}\}$. We have given the CNF-defense-formulas in Example 3. $F^\text{def}_T$ thus is given by the AF in Figure 4. Considering, for instance, argument $c$, where $C^T_c = \{\{a\}, \{b, d\}\}$, one can see that in $F^\text{def}_T$ it takes $a$ or $b$ in order to defend $c$ from $\alpha_{c,\{a,b\}}$, and $b$ or $d$ in order to defend $c$ from $\alpha_{c,\{b,d\}}$.

**Proposition 8.** For each adm-closed extension-set $S$ where $\emptyset \in S$ it holds that $\text{adm}(F^\text{def}_S) = S$. 

![Fig. 4. AF $F^\text{def}$ used in Example 4](image-url)
Proof. By Proposition 7, \( S \subseteq \text{adm}(F^\text{def}_S) \) holds for every extension-set \( S \).
It remains to show \( \text{adm}(F^\text{def}_S) \subseteq S \). Consider some \( S \in \text{adm}(F^\text{def}_S) \). First of all, \( S \) cannot contain any of the self-attacking arguments \( \alpha_{a,\gamma} \). For \( S = \emptyset \), \( S \in S \) by definition. If \( S \) consists of exactly one argument, i.e. \( S = \{a\} \), it must hold that \( \forall b \in A \ s.t. \ b \mapsto a : \ a \mapsto b \). For that, \( C_a^S = \{\} \) must hold, therefore \( S \subseteq S \). Now assume \( S \) contains at least two arguments. \( S \) being conflict-free, by construction of \( R^f_s \), guarantees that \( \forall a, b \in S : (a, b) \in \text{Pairs}_S \). Let \( s \in S \) with \( \{s\} \notin \text{adm}(F^\text{def}_S) \). Then we have \( \alpha_{s,\gamma} \mapsto s \) for each \( \gamma \in C_s^S \). Since \( s \) is defended by \( S \), for each \( \gamma \in C_s^S \), \( \exists t_\gamma \in (S \setminus \{s\}) : t_\gamma \mapsto \alpha_{s,\gamma} \). By definition of \( F^f_s \), \( t_\gamma \) occurs in the clause \( \gamma \). It follows that \( T = \{t_\gamma \mid c \in C_s^S\} \) is a model of \( C_s^S \) and thus also of the defense-formula \( D_s^S \). Then by Lemma 6 there is some \( S'_s \subseteq T \cup \{s\} \) (note that also \( S'_s \subseteq S \)) with \( s \in S'_s \) such that \( S' \in S \). Recall also that in case \( \{s\} \in \text{adm}(F^\text{def}_S) \), we know from above that \( \{s\} \in S \) (say \( S'_s = \{s\} \)). Knowing that \( \forall a, b \in S : (a, b) \in \text{Pairs}_S \), since \( S \) is adm-closed we get \( S'_{s_1} \cup S'_{s_2} \in S \) for any \( s_1, s_2 \in S \). Hence \( S \subseteq S \). \( \Box \)

Lemma 7. For each incomparable extension-set \( S \), it holds that \( S \) is pref-closed iff \( S \cup \{\emptyset\} \) is adm-closed.

Proof. Follows immediately from Lemma 4, the fact that \( \text{Pairs}_S = \text{Pairs}_{S \cup \{\emptyset\}} \) for all \( S \subseteq 2^A \), and Definition 4. \( \Box \)

Proposition 9. For each non-empty, pref-closed extension set \( S \) it holds that \( \text{pref}(F^\text{def}_S) = S \).

Proof. Let \( S' = S \cup \{\emptyset\} \). By construction, \( F^\text{def}_{S'} = F^\text{def}_S \). From Lemma 7 and Proposition 8 we thus obtain \( \text{adm}(F^\text{def}_{S'}) = S' \). As preferred extensions are \( \subseteq \)-maximal admissible sets and since \( S \) is incomparable (Lemma 4), \( \text{pref}(F^\text{def}_S) = S \). \( \Box \)

This result together with the fact that for each \( \text{AF} F' \) there is an \( \text{AF} F \) such that \( \text{pref}(F') = \text{sem}(F) \) (see [17]), yields the following result.

Proposition 10. For each non-empty, pref-closed extension set \( S \), there exists an \( \text{AF} F \), such that \( \text{sem}(F) = S \).

We now have all results at hand to characterize the signatures for the semantics we deal with in this paper. All relations in the subsequent theorem follow immediately from results in this section together with the corresponding characterizations given in Proposition 1–3.

Theorem 1. The signatures for the considered semantics are given by the following collections of extension sets.

\[
\begin{align*}
\Sigma_{\text{cf}} &= \{S \neq \emptyset \mid S \text{ is downward-closed and tight}\} \\
\Sigma_{\text{adm}} &= \{S \neq \emptyset \mid S \text{ is adm-closed and contains } \emptyset\} \\
\Sigma_{\text{native}} &= \{S \neq \emptyset \mid S \text{ is incomparable and } \text{acl}(S) \text{ is tight}\} \\
\Sigma_{\text{stb}} &= \{S \mid S \text{ is incomparable and tight}\} \\
\Sigma_{\text{stage}} &= \{S \neq \emptyset \mid S \text{ is incomparable and tight}\} \\
\Sigma_{\text{pref}} &= \Sigma_{\text{sem}} = \{S \neq \emptyset \mid S \text{ is pref-closed}\}
\end{align*}
\]
Theorem 2. The following relations hold:

\[ \Sigma_{\text{naive}} \subset \Sigma_{\text{stage}} \subset \Sigma_{\text{sem}} = \Sigma_{\text{pref}}, \quad \Sigma_{\text{stb}} = \Sigma_{\text{stage}} \cup \{\emptyset\} \]

\[ \{\text{dcl}(S) \mid S \in \Sigma_{\text{naive}}\} = \Sigma_{\text{cf}} \]

Proof. In what follows, we make implicit use of the results from Theorem 1. First, if an incomparable extension-set \(S\) is tight, then also \(\text{dcl}(S)\) is tight (using \(\text{Pairs}_{\text{dcl}(S)} = \text{Pairs}_{\Sigma_{\text{dcl}(S)}}\)). Thus, \(\Sigma_{\text{naive}} \subseteq \Sigma_{\text{stage}}\); \(\Sigma_{\text{naive}} \subset \Sigma_{\text{stage}}\), i.e. that the relation is proper, can been seen from the AF discussed in Example 1. Relation \(\Sigma_{\text{stage}} \subseteq \Sigma_{\text{sem}}\) follows from Lemma 3. \(\Sigma_{\text{stage}} \supseteq \Sigma_{\text{sem}}\) does not hold by Example 2, therefore \(\Sigma_{\text{stage}} \subseteq \Sigma_{\text{sem}}\). \(\Sigma_{\text{cf}} \subset \Sigma_{\text{adm}}\) follows in the same manner by Lemma 2 and the fact that \(S = \{\emptyset, \{a, b\}\}\) is adm-closed but not downward-closed and therefore \(S \in \Sigma_{\text{adm}}\), but \(S \notin \Sigma_{\text{cf}}\). The remaining relations in the second line follow from the definition of \(\text{dcl}(\cdot)\) and respectively Lemma 7. \(\square\)

5 Strict Realizability

Inspecting the proofs of Propositions 4 and 5 shows that for each extension set \(S\) that is realizable w.r.t. conflict-free sets (or naive semantics), there is an AF of the form \(F = (\text{Args}_S, R)\) (that is, without additional arguments) with the same outcome. Given a semantics \(\sigma\), let us thus call an extension set \(S \subseteq 2^A\) strictly \(\sigma\)-realizable, if there exists an AF \(F = (\text{Args}_S, R)\) such that \(\sigma(F) = S\). Next, we show that such a property does not hold for the remaining semantics.

Example 5. Consider \(S = \{\emptyset, \{a\}, \{a, b\}\}\). \(S\) is adm-closed, cf. Definition 4. Indeed for \(F = (\{a, b, c\}, \{(a, c), (c, b)\})\), we have \(\text{adm}(F) = S\), thus \(S \in \Sigma_{\text{adm}}\). However, there does not exist an \(F' = (A, R)\) with \(\sigma(F') = S\) and \(A = \{a, b\}\), since by \(\{a, b\} \in S\) there cannot be any attack in \(F'\). But then \(\text{adm}(F') = \{\emptyset, \{a\}, \{b\}, \{a, b\}\} \neq S\) is obvious.

Example 6. Consider \(S = \{\{a, b\}, \{a, c, e\}, \{b, d, e\}\}\). Figure 2 shows an AF (with additional arguments) realizing \(S\) as its semi-stable, and respectively, preferred extensions. Suppose there exists an AF \(F = (\text{Args}_S, R)\) such that \(\sigma(F) = S\). Since \(\{a, c, e\}, \{b, d, e\} \in S\), it is clear that \(R\) must not contain an edge involving \(e\). But then, \(e\) is contained in each \(E \in \sigma(F)\) (for the case of semi-stable extensions, since \(e\) is not attacked in such \(F\)). It follows that \(\sigma(F) \neq S\).

The previous example does not apply to stable and stage semantics (\(S\) is not tight cf. Definition 3). In fact, we need a different, slightly more involved, argument.

Example 7. Consider the extension-set \(S = \{\{a, b, c\}, \{a, b, c'\}, \{a, b', c\}, \{a, b', c'\}, \{a', b, c\}, \{a', b, c'\}, \{a', b', c\}\}\). It is easy to verify that \(S\) is non-empty, incomparable and tight. Hence, by Proposition 6, \(S\) is stb-realizable. However the AF provided by Proposition 6 makes use of an argument not in \(\text{Args}_S = \{a, b, c, a', b', c'\}\). We now show that there is no AF \(F = (\text{Args}_S, R)\) such that \(\text{stb}(F) = S\) or \(\text{stage}(F) = S\). First, given that the sets in \(S\) must be conflict-free the only possible attacks in \(R\) are \((a, a')\),
Table 1. Complexity of the recasting problem.

<table>
<thead>
<tr>
<th></th>
<th>stb</th>
<th>stage</th>
<th>pref</th>
<th>sem</th>
</tr>
</thead>
<tbody>
<tr>
<td>stb</td>
<td>-</td>
<td>NP-c</td>
<td>NP-c</td>
<td>NP-c</td>
</tr>
<tr>
<td>stage</td>
<td>trivial</td>
<td>trivial</td>
<td>trivial</td>
<td>trivial</td>
</tr>
<tr>
<td>pref</td>
<td>$\Pi^P_2$-c</td>
<td>$\Pi^P_2$-c</td>
<td>-</td>
<td>trivial</td>
</tr>
<tr>
<td>sem</td>
<td>$\Pi^P_2$-c</td>
<td>$\Pi^P_2$-c</td>
<td>trivial</td>
<td>-</td>
</tr>
</tbody>
</table>

(a', a), (b, b'), (b', b), (c, c'), (c', c). We next argue that all of them must be in $R$. First consider the case of $stb$. As $\{a, b, c\} \in stb(F)$ it attacks $a'$ and the only chance to do so is $(a, a') \in R$ and similar as $\{a', b, c\} \in stb(F)$ it attacks $a$ and the only chance to do so is $(a', a) \in R$. By symmetry we obtain $\{(b, b'), (b', b), (c, c'), (c', c)\} \subseteq R$. Now let us consider the case of stage. As $\{a, b, c\} \in stage(F) \subseteq naive(F)$ either $(a, a') \in R$ or $(a', a) \in R$. Consider $(a, a') \notin R$ then $\{(b, b'), (b', b), (c, c'), (c', c)\} \notin S$. However, for the resulting framework $F = (A, R)$, we have that $\{a', b', c'\} \in stb(F) = stage(F)$, but $\{a', b', c'\} \notin S$.

6 Complexity

In this section we exploit our results to give a preliminary complexity analysis in terms of the problem of recasting: given an AF $F_1 \in AF_A$ and semantics $\sigma_1$ and $\sigma_2$, decide whether there exists an $F_2 \in AF_A$, such that $\sigma_1(F_1) = \sigma_2(F_2)$. By the very nature of signatures, this is equivalent to test $\sigma_1(F_1) \in \Sigma_{\sigma_2}$. Table 1 shows our results: an entry gives the complexity of deciding whether $\sigma_1(F_1) \in \Sigma_{\sigma_2}$. $\Pi^P_2$-c abbreviates completeness for class $\Pi$, "trivial" means that each instance is a "Yes"-instance.

**Theorem 3.** The complexity results depicted in Table 1 hold.

**Proof (Sketch).** The "trivial" results are immediate by Theorem 2. Further using that $\Sigma_{stb} = \Sigma_{stage} \cup \{\emptyset\}$ and $\Sigma_{stage} \subset \Sigma_{sem} = \Sigma_{pref}$ we have that $stb(F) \in \Sigma_{\sigma}$ ($\sigma \in \{stage, pref, sem\}$) iff $stb(F) \neq \emptyset$. Deciding whether an AF has a stable extension is well-known to be NP-complete.

Finally, we consider the $\Pi^P_2$-entries, i.e. $\sigma_1 \in \{pref, sem\}$, $\sigma_2 \in \{stb, stage\}$. Since $\sigma_1(F) \neq \emptyset$ for any AF $F$, and $\Sigma_{stb} = \Sigma_{stage} \cup \{\emptyset\}$, we can stick to $\sigma_2 = stb$. Membership is by an algorithm disproving, given an $F = (A, R)$, $\sigma_1(F) \in \Sigma_{stb}$: guess sets $B \subseteq A$, $\{A_s \subseteq A \mid s \in B\}$ and $a \in A \setminus B$; use an NP-oracle to check $B \in \sigma_1(F)$ [14, 16]; for all $s \in B$ check $A_s \in adm(F)$, $\{a, s\} \subseteq A_s$. Intuitively, the algorithm accepts (i.e. all checks holds), if $B \in \sigma_1(F)$ violates tightness for $\sigma_1(F)$.

We show $\Pi^P_2$-hardness for $\sigma_1 = pref$ (as $pref$ semantics can be efficiently reduced to $sem$ semantics [17], the result for $\sigma_1 = sem$ follows): Given QBF $\Phi = \forall Y \exists Z \varphi(Y, Z)$, where $\varphi$ is a CNF $\bigwedge_{c \in C} c$ with each $c \in C$ a disjunction of literals from $X = Y \cup Z$, let $F_\varphi = (A_\varphi, R_\varphi)$ with $A_\varphi = \{\varphi, g\} \cup C \cup X \cup \bar{X} \cup \{a, b, c, d, e, f\}$ and
\[ R_\Phi = \{\langle c, \varphi \rangle \mid c \in C\} \cup \{\langle x, \bar{x} \rangle \mid x \in X\} \cup \{\langle \varphi, g \rangle, \langle g, g \rangle \} \cup \{(g, z), (g, \bar{z}) \mid z \in Z\} \cup \{(a, d), (d, a), (b, c), (c, b), (c, d), (d, c), (c, f), (d, f), (f, e), (f, f), (\varphi, f)\} \]

We illustrate \( F_\Phi \) for the QBF \( \Phi = \forall y_1, y_2 \exists z_3, z_4 (\left( y_1 \lor y_2 \lor z_3 \right) \land \left( y_2 \lor \neg z_3 \lor \neg z_4 \right) \land (y_2 \lor z_3 \lor z_4)) \).

\[ F_\Phi \text{ links the reduction from } [14] \text{ with the AF from Figure } 2 \text{ via } \varphi \mapsto f. \text{ One can show that } \text{pref}(F_\Phi) \in \Sigma_{stb} \text{ iff } \varphi \text{ is contained in each } E \in \text{pref}(G_\Phi). \]

7 Discussion

In this work, we initiated a study on the characteristics the set of extensions w.r.t. a given semantics satisfy. For the semantics naive, stable, stage, preferred, and semi-stable we have an exact picture fully describing their signatures \( \Sigma_\sigma \). These results also tell about the limits of global disagreement (a notion introduced in [8]) that can be modelled within AFs, e.g. our results show that preferred and semi-stable semantics are able to express more disagreement than stage semantics: \( \Sigma_{\text{stage}} \subset \Sigma_{\text{pref}} = \Sigma_{\text{sem}}. \)

We have also touched the concept of strict realizability, i.e. the question whether a set \( S \) of extensions can be realized by an AF \( F \) having no additional arguments (all arguments of \( F \) appear in \( S \)). Exact characterizations for strict signatures are important foundations for simplifications of AFs and thus a natural next step for our studies. In general, we believe that results on signatures yield useful methods for pruning the search space in algorithms for abstract argumentation.

Further directions of future work are an investigation of other important semantics, in particular complete [13], resolution-based grounded [2], and cf2-semantics [4], and an according extension of our complexity analysis. Finally, since we have viewed semantics here only in an extension-based way, it would also be of high interest to extend our studies to labelling-based semantics [11].

References


Secrecy preserving BDI Agents based on Answer Set Programming

Patrick Krümpelmann and Gabriele Kern-Isberner
Technische Universität Dortmund, Information Engineering Group

Abstract. We consider secrecy from the point of view of an autonomous knowledge-based and resource-bound agent with incomplete and uncertain information, situated in a multiagent system. We investigate properties of secrecy and the preservation thereof in this setting and formalize resulting desirable properties. We show that a notion of secrecy is dependent on multiple factors such as the agents way of information processing and change, the modeling of other agents information and their reasoning capabilities as well as the agents way of acting. We make these factors explicit and formalize them. Based on these ideas we develop a general epistemic agent model with its secrecy relevant components. To illustrate and exemplify our approach we use a BDI-based agent model instance and answer set programming for knowledge representation. We implemented our developed extendable framework for secrecy-preserving agents based on JAVA and answer set programming.

1 Introduction

Secrecy in multiagent systems is generally imposed by some global definition of secret information from a global, complete and certain view of the entire system which aim at strong notions of secrecy of a whole (multiagent) system, for an overview see [7,16]. Moreover, while substantial work on these definitions of secrecy exists mechanisms for secrecy preservation in multiagent systems are lacking. In this work we consider secrecy and secrecy preservation from the local point of view of an autonomous knowledge-based agent with incomplete and uncertain information, situated in a multiagent system1.

Agents reason under uncertainty about the state of the environment, the reasoning of other agents and possible courses of action. They pursue their goals by performing actions in the environment including the communication with other agents. On the one hand, the exchange of information with other agents is often essential for an agent in order to achieve its goals; especially if the agent is part of a coalition. On the other hand the agent is interested, or obliged, not to reveal certain information, its secrets. Restriction of communication leads to a loss of performance and utility of the individual agent, coalitions and the whole multiagent system. A good solution of the implied conflict between the agent’s

1 This article is a slightly extended version of [12].
goal to preserve secrecy and its other goals is one that restricts communication as little as necessary in order to preserve secrecy.

Secrecy of information and in particular the inference problem depend on the representation of information and the appropriate modeling of background information and of the reasoning capabilities of the agents.

Our contributions lay in several aspects. We investigate, motivate and formalize novel general properties of an agent model with respect to secrecy and secrecy preservation from a subjective perspective of an agent with incomplete information. We develop an epistemic agent model for secrecy preservation, which is based on the abstract model presented in [11]. We show that besides the pure declaration of secrets, the properties of the belief change, the attacker modeling and the means-end reasoning components of the agent are essential for secrecy declaration and preservation. and define the properties of each of the three components in detail. Moreover, we specify concrete instances which are based on answer set programming (ASP) [6] to illustrate how the properties can be satisfied. We implemented the general framework as well as the ASP instance presented in this work using JAVA and available ASP solvers [4,5]. ASP is used as an efficient method for non-monotonic reasoning of the agents as well as to represent meta-level reasoning. The properties we formalize are general and shed a bit more light on the complexity of this problem. Moreover, our model and implementation can be used for simulations and evaluation of different instantiations of the framework.

The remainder of this paper is structured as follows. First we give a very brief introduction to ASP in Section 2. Then, in Section 3, we motivate and informally develop desiderata of a secrecy preserving agent based on the belief change, the attacker modeling and the means-end-reasoning component of an agent. Based on these ideas we formalize our notion of an epistemic agent in Section 4. In Section 5 we elaborate the first, the belief change, component of an agent with respect to secrecy. In Section 6 we elaborate the second component by presenting a formalization and an approach to attacker modeling and its relevance for secrecy preservation. In Section 7 we consider the third component and develop properties and for means-end-reasoning and how to satisfy them in our instance. In Section 8 we sum up, discuss the relation to other approaches and give an outlook.

2 Answer set Programming Basics

We give a brief introduction to answer set programming [6]. Let $At$ be the set of all atoms and $Lit$ the set of all literals $Lit = At \cup \{\neg A \mid A \in At\}$. A rule $r$ is written as $H(r) \leftarrow B^+(r), B^-(r)$, the head of the rule $H(r)$ is either empty or consists of a single literal, the body consists of $B^+ = \{L_1, \ldots, L_m\}$ and $B^- = \{\text{not } L_{m+1}, \ldots, \text{not } L_n\}$ with $L_1, \ldots, L_n \in Lit$. The language of rules constructed over the set of atoms $At$ is referred to as $L_{Asp}^{At}$. A finite set of rules from $L_{Asp}^{At}$ is called an extended logic program $P \subseteq L_{Asp}^{At}$. A state $S$ is a set of literals that does not contain any complementary literals $L$ and $\neg L$. A state $S$
is a model of a program $P$ if for all $r \in P$ if $\mathcal{B}(r)^+ \subseteq S$ and $\mathcal{B}(r)^- \cap S = \emptyset$ then $H(r) \cap S \neq \emptyset$. The reduct $P^S$ of a program $P$ relative to a set $S$ of literals is defined as

$$P^S = \{ H(r) \leftarrow \mathcal{B}(r)^+ \mid r \in P, \mathcal{B}(r)^- \cap S = \emptyset \}.$$ 

An answer set of a program $P$ is a state $S$ that is a minimal model of $P^S$. The set of all answer sets of $P$ is denoted by $AS(P)$. Rule schemas can use variables which we denote by $x, y, z$ and $\_\_$ for the anonymous variable [6].

### 3 Properties of Secrecy and Secrecy Preservation

In this section we argue that the definition of secrecy is complex and dependent on various aspects which influence the actually obtained secrecy and the restriction of information flow. Furthermore, we elaborate the key ideas and properties of secrecy preserving agents. In the following we give the introduction to our running example and then statements about secrecy followed by examples.

**Example 1.** Consider an employee, $emp$, working in a company for his boss $boss$. He wants to attend a strike committee meeting ($scm$) next week and has to ask his boss for a day off in order to attend. It is general knowledge that the agent $boss$ puts every agent who attends the $scm$ on her blacklist of employees to be fired next.

Secrets are not uniform in their content as an agent has different secrets with respect to different agents.

**Example 2.** In our example, $emp$ wants to keep his attendance to the $scm$ secret from $boss$ but not from other employees that also want to attend the $scm$.

Secrets are also not uniform with respect to their strength. That is, an agent wants to keep some information more secret than other. These differences in strength of secrets arise naturally from the value of the secret information. The value of secret information depends on the severeness of the negative effects, or the cost, for the agent resulting from disclosure of the secret information. These costs can differ widely and consequently the agent is interested in not revealing secret information to different degrees.

**Example 3.** $emp$ does not even want his $boss$ to be suspicious about him attending the $scm$ (secrecy with respect to a credulous reasoner). He also does not want other employees that are against the strike to know that he attends the $scm$. However, with respect to the latter he considers it sufficient that they do not know for sure that he attends (secrecy with respect to a skeptical reasoner).

Secrets are also not static, they arise, change and disappear during runtime of an agent such that it has to be able to handle these changes adequately.

**Example 4.** If $emp$ realizes that $boss$ overheard his phone call with the strike committee he should give up his corresponding secret.
These considerations lead to the following formulation of properties of secrets:

- **S1** secrets can be held with respect to specific agents
- **S2** secrets can vary in strength
- **S3** secrets can change over time

Now we want to formulate properties of a secrecy preserving agent and begin with an informal formulation. We assume a multiagent system with a set of agents $\mathcal{A}$. We use the agent identifier $X$ to denote an arbitrary agent. For the representation of the secrecy scenario it is convenient to focus on the communication between two agents, the modeled agent $\mathcal{D}$ which wants to defend its secrets from a potentially attacking agent $\mathcal{A}$. Defining secrets does not define the preservation of secrecy and its properties. The intuitive formulation of our notion of secrecy preservation can be formulated as: *An agent $\mathcal{D}$ preserves secrecy if, from its point of view, none of its secrets $\Phi$ that it wants to hide from agent $\mathcal{A}$ is, from $\mathcal{D}$’s perspective, believed by $\mathcal{A}$ after any of $\mathcal{D}$’s actions (given that $\mathcal{A}$ does not believe $\Phi$ already).*

The actual quality of secrecy preservation is highly dependent on the accuracy of the view of $\mathcal{D}$ on the agent $\mathcal{A}$ and its supposed reasoning capabilities as well as on $\mathcal{D}$’s information processing and adaptation of its beliefs and view on $\mathcal{A}$ in the dynamic scenario. To make the importance of the attacker model clear, a completely ignorant agent could never subjectively violate secrecy as it would ignore its violation of secrecy. Likewise underestimating as well as overestimating the capabilities of an agent $\mathcal{A}$ can lead to a violation of secrecy. In particular a secrecy preserving agent should satisfy the following properties:

- **P1** The agent is aware of the information communicated to other agents and the meta-information conveyed by its actions,
- **P2** The agent simulates the reasoning of other agents,
- **P3** The agent considers possible meta-inferences from conspicuous behavior such as (a) self-contradiction, (b) refusal,
- **P4** For all possible states and perceptions the agent does not perform any action that leads to secrecy violation,
- **P5** The agent only weakens secrets if it is unavoidable due to information coming from third parties and only as much as necessary.

As we shall see, the properties (P1) and (P5) are related to the belief change component of $\mathcal{D}$, (P2) and (P3) to the way $\mathcal{D}$ models $\mathcal{A}$ and (P4) to the means-end reasoning behavior of $\mathcal{D}$. In the following we elaborate on all properties, formalize them and develop corresponding agent components and show which formalized properties are satisfied.

### 4 Formal Framework

We present an epistemic model of agency which stresses the knowledge representation and reasoning under uncertainty and incorporates secrets, and views of an agent on the information available to other agents. The reasoning under uncertainty is formalized by belief operators which differ in their credulity. We then use these notions to define secrets and secrecy preservation.
The general framework as presented in [11] generalizes a variety of agent models. Here, we use a more concrete model loosely based on the well known BDI (Beliefs, Desires, Intentions) architecture [3,15]. Note that the BDI model just serves as an example agent model and that all properties and operators developed here are independent of it and are applicable to virtually all agent models. In our epistemic view of agency, the agent’s epistemic state contains a representation of its current desires and intentions which guides its behavior. The functional component of a BDI agent consists of a change operation of the epistemic state and an action function, executing the next action as determined by the current epistemic state. Our agent model is illustrated in Figure 1.

**Definition 1 (Epistemic BDI Agent).** An agent $\mathcal{D}$ is a tuple $(\mathcal{K}_D, \xi_D)$ comprising an epistemic state $\mathcal{K}_D$ and a functional component $\xi_D$. A BDI-Epistemic-State is a tuple $\mathcal{K}_D = \langle V_{\mathcal{D}}, V_{\mathcal{D}}, S_{\mathcal{D}}, \Delta_{\mathcal{D}}, I_{\mathcal{D}} \rangle$. It consists of a world view $V_{\mathcal{D}, W}$, a set of agent views $V_{\mathcal{D}} = \{ V_{\mathcal{D}, X} \mid X \subseteq \mathcal{A} \setminus \{ D \} \}$, a set of secrets $S_{\mathcal{D}}$, a set of desires $\Delta_{\mathcal{D}}$, and a set of intentions $I_{\mathcal{D}}$. We refer to the first component of a BDI-epistemic-state as the agent’s beliefs $\mathcal{B}(\mathcal{K}_D) = B_{\mathcal{D}}$. We set $V_{W}(\mathcal{B}) = V_{W}(\mathcal{K}_D) = V_{\mathcal{D}, W} \subseteq \mathcal{L}_{asp}^{\mathcal{A}_{\mathcal{D}}}$. $V_{X}(\mathcal{B}) = V_{X}(\mathcal{K}_D) = V_{\mathcal{D}, X} \subseteq \mathcal{L}_{asp}^{\mathcal{A}_{\mathcal{D}}}$, and $S(\mathcal{K}_D) = S(\mathcal{B}) = S_{\mathcal{D}}$. The functional component $\xi_D = (\circ_D, \text{act}_D)$ consists of a change operator $\circ_D$ and an action operator $\text{act}_D$.

A belief operator determines the currently held beliefs of the agent given a view. In the ASP setting beliefs are represented by an answer set, i.e. a set of literals, and a view by an extended logic program. An agent with incomplete and uncertain information might employ different belief operators which are more or less credulous. A belief operator is more credulous than another one if for all views the belief set of the latter is a subset of the belief set of the former.

**Definition 2 (Belief Operators).** A belief operator is a function $\text{Bel}: 2^{\mathcal{L}_{asp}^{\mathcal{A}_{\mathcal{D}}}} \rightarrow 2^{\text{Lit}}$. A finite family of belief operators is denoted by $\Xi$ and always contains the
ignorant operator $\text{Bel}_0(V) = \emptyset$. We assume a credulity order $\prec$ on $\Xi$ such that if $\text{Bel} \prec \text{Bel}'$ for some $\text{Bel}, \text{Bel}' \in \Xi$ then for all $V \in \mathcal{L}_{At}^{\text{asp}}, \text{Bel}(V) \subseteq \text{Bel}'(V)$.

The ASP belief operator family is given by $\Xi^{\text{asp}} = \{\text{Bel}_{\text{skep}}^{\text{asp}}, \text{Bel}_{\text{cred}}^{\text{asp}}, \text{Bel}_0\}$, $\text{Bel}_{\text{cred}}^{\text{asp}}(P) = \cap \mathcal{A}(P)$ and $\text{Bel}_{\text{skep}}^{\text{asp}}(P) = \cup \mathcal{A}(P)$ and $\text{Bel}_{\text{cred}}^{\text{asp}} \succ \text{Bel}_{\text{skep}}^{\text{asp}} \succ \text{Bel}_0$.

The definition of a family of belief operators abstracts from the underlying formalism and inference mechanism. Thereby it captures a wide range of formalisms from purely qualitative ones to plausibilistic ones. The ASP instance considered here is just one example, used for the illustration of the approach in this paper.

**Note:**

**Definition 3 (Belief Operators).** A belief operator is a function $\text{Bel} : \mathcal{L}_{At}^{\text{asp}} \rightarrow \text{Lit}$ such that for $V$, $\text{Bel}(V) \subseteq \text{Lit}$. $\Xi$ is a finite family of belief operators $\Xi$ plus the ignorant operator $\text{Bel}_0(V) = \emptyset$. A unique credulity-value is assigned to each operator by the injective function $s : \Xi \rightarrow [0, 1] \cup \{s_0\}, s_0 < 0$, such that if $s(\text{Bel}) < s(\text{Bel}')$ for some $\text{Bel}, \text{Bel}' \in \Xi$ then for all $V \in \mathcal{L}_{At}^{\text{asp}}, \text{Bel}(V) \subseteq \text{Bel}'(V)$.

The ASP belief operator family is given by $\Xi^{\text{asp}} = \{\text{Bel}_{\text{skep}}^{\text{asp}}, \text{Bel}_{\text{cred}}^{\text{asp}}, \text{Bel}_0\}$, $\text{Bel}_{\text{cred}}^{\text{asp}}(P) = \cap \mathcal{A}$ and $\text{Bel}_{\text{skep}}^{\text{asp}}(P) = \cup \mathcal{A}$ and $s(\text{Bel}_{\text{cred}}^{\text{asp}}) = 1, s(\text{Bel}_{\text{skep}}^{\text{asp}}) = 0, s(\text{Bel}_0) = s_0$.

To define secrets, the information to be kept secret has to be defined. Also, the agent from which the information shall be kept secret has to be defined and lastly the strength of the secret has to be expressed. We make use of the belief operators to express the strength of a secret.

**Definition 4 (Secrets).** A secret is a tuple $(\Phi, \text{Bel}, \mathcal{A})$ which consists of a formula $\Phi \in \text{Lit}$, a belief operator $\text{Bel} \in \Xi$ and an agent identifier $\mathcal{A} \in \mathfrak{A}$. The set of secrets of agent $\mathcal{D}$ is denoted by $S(\mathcal{K}_\mathcal{D})$.

Assigning a more credulous belief operator to a secret leads to a stronger protection of secret information, as illustrated in Example 3. That is, if $\mathcal{D}$ reveals some information, a credulous attacker might infer some secret information while a skeptical one with the same revealed information might not. In the former case the defender should not have revealed the information. Formally, considering two secrets $(\Phi, \text{Bel}, \mathcal{A})$ and $(\Phi, \text{Bel}', \mathcal{A})$, the former is stronger than the latter iff $\text{Bel} \succ \text{Bel}'$.

**Observation 1** The definition of secrets in Definition 4 satisfies (S1), (S2) and (P2).

**Example 5.** We model the $\text{scm}$ scenario from Example 2 and in particular the initial epistemic state of the employee $\text{emp}$, $\mathcal{K}_\text{emp} = \langle \{\text{emp}, W, \text{emp}, \mathcal{S}_\text{emp}\}, \Delta_\text{emp}, \mathcal{I}_\text{emp} \rangle$, with $\mathcal{V}_\text{emp} = \{\text{emp}, \text{boss}\}$. We assume that $\text{emp}$ and $\text{boss}$ share the same background knowledge, such that $\mathcal{V}_{\text{emp}, W} = \mathcal{V}_{\text{emp}, \text{boss}} = P_\text{view}$ with:
The program encodes that emp has to be excused in order to not go to work (r₁). He is excused if the attends the scm or if he has a medical appointment (r₁–r₂). If he asks to be excused these two possible explanations exist (r₄–r₅). If he is absent without being excused he will be blacklisted (r₆). If he attends the scm, and is thus excused, he will still be blacklisted (r₇). He normally goes to work (r₈). The set of answer sets is \( \text{AS}(P_{\text{view}}) = \{\{\text{attend}_\text{work}\}\} \). The secret of the employee is \( \text{S}_{\text{emp}} = \{\{\text{attend}_\text{scm}, \text{Bel}^{\text{asp}}, \text{boss}\}\} \). The initial set of desires of the employee is \( \Delta_{\text{emp}} = \{\text{attend}_\text{scm}\} \).

For secrecy preservation the dynamics of the epistemic state induced by actions and perceptions have to be considered. We assume a set of possible actions \( \text{actions} \) and a set of possible perceptions \( \text{percepts} \), including the empty ones. To make the formalism more comprehensible and to illustrate a concrete instance we consider communicating agents here. Note that other types of actions and perceptions, such as manipulations in some environment are also captured by the general framework. For the illustration here, we assume that actions as well as perceptions \( \tau \) are speech acts from a set of speech acts \( \langle A_s, \{A_{s_1}, \ldots, A_{s_n}\}, \text{type}, \Phi \rangle \) specifying the source \( A_s \in \mathfrak{A} \), the receivers \( A_{s_1} \in \mathfrak{A} \) to \( A_{s_n} \in \mathfrak{A} \), the type \( \text{type} \) and the informational content \( \Phi \in \text{Lit} \). The main difference between perceptions and actions is that perceptions represent actions performed by other agents while actions represent the actions the agent under consideration has performed. We differentiate between requesting speech acts \( \Psi_R = \{\text{query}, \text{justify}\} \) and informative speech acts \( \Psi_I = \{\text{inform}, \text{answer}, \text{justification}\} \), so type \( \in \Psi_R \cup \Psi_I \). The set of all possible speech acts is denoted by \( \Gamma = \text{percepts} = \text{actions} \). For each perception \( p \in \text{percepts} \) an agent cycle results in a new epistemic state determined by \( \mathcal{K}_D \circ_D p \circ_D \text{act}_D (\mathcal{K}_D \circ_D p) \). The set of all possible successive epistemic states of agent \( D \) is determined by the set of initial epistemic states \( A_0^D \) and all respective successor states for all possible courses of perceptions and corresponding actions of \( D \), i.e. \( \Omega_{\text{act}_D, \circ_D}(A_0^D, \text{percepts}) = \{\mathcal{K} | \mathcal{K} = \mathcal{K}_0 \circ_D p_0 \circ_D \text{act}_D (\mathcal{K}_0 \circ_D p_0) \circ_D \ldots \circ_D p_0 \ldots, p_i \in \text{percepts}, i \in \mathbb{N}_0, \mathcal{K}_0 \in A_0^D \} \).

Our intuitive idea of secrecy preservation as given in Section 3 expresses that we want to assure that a secrecy preserving agent always maintains an epistemic state in which it believes that no other agent believes in something that it wants to keep secret. More exactly, it also distinguishes between secrets towards different agents and what it means to it that the information is kept secret. The term “always maintains” means that for all possible scenarios of communication the agent acts such that a safe epistemic state is maintained.
Definition 5 (Secrecy-preserving Agent). Let $D = (KD, (\circ_D, \text{act}_D))$ be an agent and percepts a set of perceptions. An epistemic state $KD$ is safe iff $\Phi \not\in \text{Bel}(V_A(KD))$ for all $(\Phi, \text{Bel}, A) \in S(KD)$.

Let $A^0_D$ be a set of initial safe epistemic states. We call $D$ secrecy preserving with respect to $A^0_D$ and percepts if and only if for all $KD \in \Omega_{\text{act}, \circ}(A^0_D, \text{percepts})$ it holds that $KD$ is safe.

Example 6. We continue the previous example and check whether the initial $K_{\text{emp}}$ is safe. The set of answer sets of $P_{\text{view}}$ is $\text{AS}(P_{\text{view}}) = \{\{\text{attend\_work}\}\}$. Consequently $\text{attend\_scm} \not\in \text{Bel}_{\text{skep}}^\text{asp}(P_{\text{view}})$ and $K_{\text{emp}}$ is safe.

Observation 2 The definition of a secrecy preserving agent in Definition 5 satisfies (P4).

We just defined the notion of a secrecy preserving agent. However, as discussed in Section 3 the actual resulting properties of secrecy preservation result from the properties of the change operation $\circ$ and the attacker modeling. Moreover, the actual preservation of secrecy is realized by the means-end-reasoning of the agent. We elaborate these aspects in the next sections.

5 Belief Change and Secrecy

We decompose the belief change of an epistemic state into sub-operations on its components. Based on these we motivate and define properties with respect to secrecy preservation which formalize and concretize the ideas given in (P1) and (P5). Finally we give concrete instances of such operators for the ASP instance and show that they satisfy the defined properties.

5.1 Structure and properties of the change operator

The change operator updates epistemic state of an agent upon incoming perceptions and actions. Formally, $K \circ \tau = K' = \langle B', \Delta', I' \rangle$. The change operator can be structured into several sub-operations for the different components of the epistemic state. Hereby the belief component is the only one being directly influenced by the new information. Therefore the change of the desires is only dependent on the changed beliefs and the update of the intentions on the changed beliefs and desires. Formally the sub-operations are $\circ_B : B \times I \rightarrow B$, $\circ_\Delta : \Delta \times B \rightarrow \Delta$ and $\circ_I : I \times B \times \Delta \rightarrow I$. The update operations can then be represented as

$$\langle B, \Delta, I \rangle \circ \tau = \langle B \circ_B \tau, \Delta \circ_\Delta (B \circ_B \tau), \circ_I(I, B \circ_B \tau, \Delta \circ_\Delta (B \circ_B \tau)) \rangle.$$ 

In this section we focus on the belief change operation and its relevance to secrecy and elaborate the desire and intention change in the context of means-end-reasoning later on. The input speech act $\tau$ for the $\circ_B$ operation can be either a perception or an action. In both cases it might be an informative or a requesting speech act. All four cases have different semantics and lead to different changes.
That is, the input has to be interpreted and represented in the language of the respective belief component. To this end we introduce translation operators \( t_W : \Gamma \rightarrow \mathcal{L}_{At}^{asp} \) for the world view and \( t_V : \mathfrak{A} \times \Gamma \rightarrow \mathcal{L}_{At}^{asp} \) for agent views. The result of the translation is then used to update the respective component by use of an inner revision operator \( * : 2\mathcal{L}_{At}^{asp} \rightarrow 2\mathcal{L}_{At}^{asp} \). With this operator the set of agent-views is revised as follows:

\[
V \ast t_V \tau = \{ V_{D,X} \ast t_V(X, \tau) \mid V_{D,X} \in \mathcal{V} \}
\]

Secrets are updated on the basis of the agent’s updated beliefs and views such that the change operator for secrets \( \ast_S \) is dependent on these as well as on the incoming information, i.e.

\[
\ast_S : 2\mathcal{L}_S \times \mathcal{L}_{At}^{asp} \times 2\mathcal{L}_{At}^{asp} \times \Gamma \rightarrow \mathcal{L}_S .
\]

We define the changes of the \( \circ_B \) operator to the components by suboperations, such that

\[
(V_W, V, \mathcal{S}) \circ_B \tau = (V_W \ast t_W(\tau), V \ast t_V \tau, \ast_S(\mathcal{S}, V_W \ast t_W(\tau), V \ast t_V \tau, \tau)) \quad (*)
\]

We define a set of properties on the just defined operations which formalize the properties (S3), (P1) and (P5). For secrecy preservation it is necessary that the agent does not give up any secrets upon reflecting its own actions since it would be able to perform arbitrary actions without violating secrecy by abandoning its secrets. Thus, the agent must not be able to preserve a safe epistemic state by modifying its secrets.

**Secrets-Invariance**, \( \circ_B \)  If \( \tau \in \text{actions} \) then \( S(B \circ_B \tau) = S(B) \)

The *Secrets-Invariance* property is restricted to inputs that are actions. These actions are those of the agent itself and perceptions reflect changes in the environment or actions of other agents. For the latter the postulate should not hold. That is, an agent should not be able to ignore the fact that a secret has been revealed due to changes in the environment or actions of other agents. This is expressed in the following property.

**Acknowledgment**, \( \circ_B \)  If \( \tau \in \text{percepts} \) then \( B \circ_B \tau \) is safe.

The changes to the set of secrets in order to achieve a safe epistemic state should be minimal. In particular, no secret that is not violated should not be changed, if it is violated and it should be weakened minimally.

**Min-Secrecy-Weakening**, \( \circ_B \)  If \((\Phi, Bel, A) \in S(B)\) and \((\Phi, Bel', A) \in S(B \circ_B \tau)\) with \( Bel \neq Bel' \) then \( \Phi \in Bel(V_A(K_D \circ_B \tau)) \) and there is no \( Bel'' \) such that \( \Phi \notin Bel''(V_A(K_D \circ_B \tau)) \) with \( Bel'' > Bel' \).

Another secrecy relevant property of belief change arises from the changes to views of other agents. An agent should not be able to preserve secrecy by ignoring the effects of its own actions on the beliefs of potentially attacking agents. In particular the information for some agent \( A \) contained in an action of \( D \) should be incorporated into \( D \)'s view on \( A \). This is formulated by the next property.
Awareness$_{oB}$ If $\tau \in \text{actions}$ then $t_W(\tau) \in V_W(B \circ_B \tau)$ and for each $A \in \mathfrak{A}$ $t_V(A, \tau) \in V_A(B \circ_B \tau)$. There might very well be actions which are not visible to all agents and therefore should also not affect the view on all agents. This is achieved by use of appropriate translation operators which select the relevant information for each agent. For agents that are not affected by the information the transformation function returns the empty set.

Observation 3 The satisfaction of Secrets-Invariance$_{o}$, Acknowledgment$_{o}$, Min-Secrecy-Weakening$_{o}$ and Awareness$_{o}$ of the belief change operator of an agent corresponds to the satisfaction of (P1), (P5) and (S3).

We consider all of the properties defined in this section essential for a secrecy preserving agent, hence the goal is to define appropriate operators.

5.2 Concrete Revision operations

In the following we define an instance of the translation operators $t_V, t_W$, the inner revision operator $*$ and the revision of secrets operator $*_{S}$.

The translation operator, in accordance with (P1), has to consider information on two levels, on the one hand the actual information, that is the informational content of the speech act. On the other hand the meta-information about the speech act that has been performed which includes especially the information about the sender and the type of speech act and information revealed by these parameters. Both aspects have to be represented in the language of a logic program. We introduce an auxiliary logic program to support the representation of information on both levels for the ASP translation operators and to enable reasoning possibilities on the meta-information which are used for attacker modeling. To this end we reify literals: for each atom $A \in \mathfrak{A}$ introduce three constant symbols $C(A) = \{a, na, \lambda a\}$ the first two to represent the literals $A, \neg A$; $\lambda a$ can stand for both occurrences $a$ and $na$. We define $\text{const}(L) = a$ if $L = A$ and $\text{const}(L) = na$ if $L = \neg A$. And $\text{var}(L) = \lambda a$ if $L = A$ or $L = \neg A$. We use a linear non-metric time ontology represented by successive time points $t$ and a literal $\text{time}(t)$ for the agent cycles. The program $P_{aux}$ consists of the following set of rules:

(1) For all $A \in \mathfrak{A}$: $A \leftarrow \text{holds}(a)$.
(2) $\neg A \leftarrow \text{holds}(na)$.
(3) $\text{related}(\lambda a, a). \text{related}(\lambda a, na). \text{at}(t) \leftarrow \text{time}(t), \neg \text{time}(s), s > t.$

The predicate related$(x, y)$ expresses that $x$ is semantically related to $y$. We make use of the auxiliary construction to represent the informational content of a speech act and formulate the translation function as follows.

Definition 6 (Translation Function). Let be $D \in \mathfrak{A}$, $\tau = \langle A_s, \{A_{r_1}, \ldots, A_{r_n}\}, type, L \rangle$ and counter = $t$. The translation functions of $D$ are defined as:
\[ t_V(\tau) = \begin{cases} \{ \text{type}(A_s, \text{const}(L), t)., L. \} & \text{if } \tau \in \Psi_I \\ \{ \text{type}(A_s, \text{var}(L), t). \} & \text{if } \tau \in \Psi_R \end{cases} \]

\[ t_W(\tau) = \begin{cases} \{ \text{type}(A_s, \text{const}(L), t)., L. \} & \text{if } A_s \neq D \text{ and } \tau \in \Psi_I \\ \{ \text{type}(A_s, \text{var}(L), t). \} & \text{else} \end{cases} \]

In general the information of a speech act is represented by the predicate \( \text{type}(A_s, \text{const}(L), t). \) with the semantics that a speech act of type \( \text{type} \) has been performed by agent \( A_s \) with logical content \( \text{const}(L) \) at time \( t \). For requesting speech acts \( \text{var}(L) \) is used to represent that information related to \( L \) has been requested. Besides this representation of the information about the speech act the logical content has to be represented. For informational speech acts this is the actual literal \( L \) of which the agent has been informed. Hence \( L \) is added to the input set for the inner revision operator for informational speech acts, unless \( D \) is updating its world view by its own action. The result of the translation operator is the input for the inner revision operator. As intended we revise a logic program by another one. For this problem a variety of operators have been developed of which we can make use. For our ASP instance we use operators satisfying a basic set of properties can be used. In particular \textit{Success}: \( P \subseteq P \cup Q \), \textit{Inclusion}: \( P \ast Q \subseteq P \cup Q \) and \textit{Vacuity}: \( \text{If } P \cup Q \text{ is consistent, then } P \cup Q \subseteq P \ast Q \) are satisfied. For details refer to, e. g., \cite{10}.

As specified by the properties given above secrets are updated only by information about actions of other agents. In this case the secrets shall be modified minimally in order to preserve secrecy. To this end, we determine the most credulous belief operator by which some information \( \Phi \) is preserved in the current view \( V \).

Formally:

\[ \text{curr}(\Xi, V, \Phi) = \max(\{ \text{Bel}' \in \Xi \mid \Phi \notin \text{Bel}'(V) \}). \]

Then we can define the change operator for secrets \( *_S \) as

\[ S(\mathcal{K}) *_{S} (V'_W, V', \tau) = \begin{cases} S(\mathcal{K}) & \text{if } A_s = D \\ \omega(S(\mathcal{K}), V') & \text{else} \end{cases} \]

with \( \omega(S(\mathcal{K}), V') = \{ (\Phi, \text{Bel}', A) \mid (\Phi, \text{Bel}, A) \in S(\mathcal{K}) \text{ and } \}

\[ \text{Bel}' = \begin{cases} \text{curr}(\Xi, V_A, \Phi) & \text{if } \text{curr}(\Xi, V_A, \Phi) \prec \text{Bel} \\ \text{Bel} & \text{else}. \end{cases} \]

If the updating information is an action of \( D \), no changes are performed. Otherwise the belief operator of any secret whose assigned operator is stronger than the currently strongest one preserving secrecy is replaced by the latter. This means that only those secrets are modified which would be violated otherwise. We can show that this specification \( *_B \) satisfies the properties postulated previously.

**Proposition 1.** Let \( D \) be an agent, \( t_V, t_W \) and \( *_S \) be operators as defined in this section and \( P_{aux} \subseteq V_A(\mathcal{K}_D) \) and \( P_{aux} \subseteq V_W(\mathcal{K}_D) \). Let \( *_B \) be an ASP base-revision operator. The \( *_B \) operator of \( D \) defined as in (*) satisfies \textit{Secrets-Invariance}_o, \textit{Acknowledgment}_o, \textit{Min-Secrecy-Weakening}_o, and \textit{Awareness}_o.
Proof. Sketch: The satisfaction of Secrets-Invariance◦, Acknowledgment◦ and Min-Secrecy-Weakening◦ follow from the definition of *S, the satisfaction of Awareness◦ follows from Definition 6 and the satisfaction of the Success postulate by *.

6 Attacker Modelling

The principles P2 (simulation) and P3 (meta-inferences) of secrecy preservation laid out in Section 3 raise the need for adequate modeling of the background information and reasoning methods and capabilities of the attacker. Both over- and underestimating the capabilities of an attacker can lead to violation of secrecy. Hence modeling these is essential for realistic preservation of secrecy. In particular information about the declaration of secrets and meta-inference from D’s behavior have to be considered. Which behavior is conspicuous and will lead to a violation of secrecy is heavily dependent on the reasoning capabilities and properties of the attacker. We define three properties of an attacker for secrecy preservation which D might take into consideration.

Secret Aware A knows which information D does not want to reveal to A if D would believe it to be true.

Contradiction Sensitive A considers self-contradictions of D with respect to information it wants to keep secret as a reason to infer the secret.

Refusal Sensitive A considers the D’s refusal to answer with respect to information it wants to keep secret as a reason to infer the secret.

We present a simple version of an ASP approach to realize views of attackers satisfying the properties. To this end we consider the following set of rules which is then used to define programs which represent a specific property and can be modularly added to a view on an A.

(1) For each (L, Bel, A) ∈ S(KD): has_secret(D, const(L)).
(2) For all A ∈ At:

contradiction(D, λa) ← inform(D, a), inform(D, na).

(3) holds(x) ← has_secret(D, x), contradiction(D, y), related(y, x).
(4) refused(D, x) ← request(λ, D, x, t1), not answer(D, y, t2), at(t2), related(x, y), t2 = t1 + 1.
(5) holds(x) ← has_secret(D, x), refused(D, y), related(y, x).

Line (1) represents the information about the secrets D has with respect to A. In (2) it is expressed that A infers that D contradicted itself with respect to an atom A if it said both A and ¬A. If D contradicted itself with respect to some secret information, then A infers that the secret holds (3). Line (4) represents that A infers that D refused to answer about x if it was requested to do so and did not inform after the request. According to (5) A infers that a secret holds if D refused to answer with respect to it. We can define programs from the defined rules and formalize the properties given above in the ASP setting.
Definition 7. Let $P_{\text{meta}}^{S-\text{aware}} = (1)$, $P_{\text{meta}}^{C-\text{sensitive}} = P_{\text{meta}}^{S-\text{aware}} \cup (2) \cup (3)$ and $P_{\text{meta}}^{R-\text{sensitive}} = P_{\text{meta}}^{S-\text{aware}} \cup (4) \cup (5)$. An attacker modeling $V_A(K_D)$ is secrecy aware if $P_{\text{meta}}^{S-\text{aware}} \subseteq V_A(K_D)$, it is contradiction sensitive if $P_{\text{meta}}^{C-\text{sensitive}} \subseteq V_A(K_D)$ and it is refusal sensitive if $P_{\text{meta}}^{R-\text{sensitive}} \subseteq V_A(K_D)$.

Observation 4 The satisfaction of the properties contradiction sensitive and refusal sensitive corresponds to the properties (P3) (a) and (b), respectively.

The determination of contradictions caused by $D$ and one of refusals can and should be more elaborate and can easily be extended and formulated by more complex logic programs, but are outside of the scope here.

7 Means-end-reasoning and Secrecy Preservation

We equipped the agent with the abilities to be aware of its secrets and to detect violation of secrecy. The question now is what are the necessary properties on the desire and intention change operators for secrecy preservation.

In any BDI system desire and intention change operators are implemented in one way or the other. Here we give a general model of intention change and show the relevant properties for secrecy preservation. Any agent has to determine how it can satisfy its intentions, high-level intentions are resolved down to atomic intentions $AtInt \subseteq I$ which can be satisfied by a single action $\alpha(I)$. In any case at some point the options to satisfy some intention have to be evaluated and one of the options has to be chosen. The options for a given intention are determined by the options function $\text{opt}: I \to 2^I$ and the evaluation results in a preference relation $<_K$ on the possible options. Here we assume that all intentions can directly be resolved to an atomic intention and set $I' := AtInt$. Then the set of maximally preferred options from which one is selected by the agent is set to $\text{pref}(\text{opt}(I)) = \max_{<_K}(\text{opt}(I))$. To show that an agent is secrecy preserving, as given by Definition 5, we have to show that it prefers secrecy preserving actions over non-secrecy preserving ones and that it always has a secrecy preserving option. Given an epistemic state $K$ and an intention $I$, an option $o \in \text{opt}(I)$ is safe iff $K \circ \alpha(o)$ is safe. Based on this definition we define a secrecy relevant property on the preference relation on options.

Confidentiality-preference For all $I \in I$, for $o, o' \in \text{opt}(I)$ in state $K$. If $o$ is safe and $o'$ is not then $o' <_K o$.

Note that this is a simplified view on an agents preferences on options that focuses on pure secrecy preservation. An agent should probably prefer “honest” options over lies and should also incorporate the utility of the options.

In combination with an $\text{opt}$ function we can show that the agent chooses secrecy preserving options, if they exist.

Lemma 1. If $<_K$ satisfies confidentiality-preference, then if there is some safe option $o \in \text{opt}(I)$, then for all $o' \in \text{pref}(\text{opt}(I))$, $o'$ is safe.
Proof. The lemma follows directly from the definitions of $\text{pref}$ and confidentiality-preference.

A preference relation satisfying confidentiality-preference alone is not sufficient to guarantee secrecy preservation since it is dependent on the existence of a safe option. Hence we define the following property of an opt function.

**Existence** For all $I \in \mathcal{I}$ and safe $K$ there exists some safe $o \in \text{opt}(I)$.

We can now formalize the dependency between the notion of a secrecy preserving agent, as in Definition 5, and the properties of option selection as defined in this section.

**Proposition 2.** Let $>$ be a preference relation on $\text{AtInt}$ and opt an options function of an agent $D$. If $>$ satisfies Confidentiality-preference and opt satisfies Existence, then $D$ is secrecy preserving.

**Proof.** Sketch: The proposition follows from Lemma 1 and the definitions of Confidentiality-preference, Existence, safe options and safe epistemic state.

8 Related work and Conclusion

We presented a theoretical, conceptional and practical account of secrecy from the subjective view of an autonomous epistemic agent. We formulated properties of secrecy and secrecy preservation and developed a framework for an ASP-based instance satisfying them. We have shown in [9] that other many aspects of notions of secrecy such as [1] and [7] can be captured by our underlying model.

To the best of our knowledge no subjective account of agent based secrecy nor a concrete model or implementation of a secrecy preserving agent system has been presented so far. The closest to this is the preliminary account of integrating query evaluation on databases [1] into an agent system, as presented in [2]. The database techniques are naturally limited to a fixed client-server architecture and to query-answer scenarios. In [2] it is proposed to use a censor to check the agents actions prior to execution and modifying them if necessary similar to the approaches from database theory for a negotiation scenario. The actual realization of this approach is left open. We argue that instead of adding an controlling instance, secrecy has to be integrated into the agents’ reasoning, deliberation and means-end reasoning processes to achieve autonomous secrecy preserving agents apt to perform well in a dynamic setting. However, secrecy is a very special epistemic goal which calls an appropriate epistemic model, operators and actions as presented in this work and can hardly be captured by the few existing approaches for maintenance goals, e.g. [8].

We see our model and implementation as a good basis for the further theoretical investigation as well as the implementation of secrecy preserving agents. It opens a plethora of possibilities for further investigation. In current work we run empirical evaluations and integrate advanced deliberation [14] and means-end reasoning techniques [13] in our model and implementation, and investigate further properties of secrecy in this model and the relation to other approaches.
Acknowledgements: This work has been supported by the DFG, Collaborative Research Center SFB876, Project A5. (http://sfb876.tu-dortmund.de)

References

OCF-networks with missing values

Gabriele Kern-Isberner and Christian Eichhorn

Department of Computer Science
Technische Universität Dortmund, Germany
gabriele.kern-isberner@cs.uni-dortmund.de
christian.eichhorn@tu-dortmund.de

Abstract. Similar to Bayesian networks, OCF-networks combine structural information encoded in a directed graph with qualitative information expressed by ranking degrees of (conditional) formulas. The benefits of such techniques are twofold: First, the high complexity of the semantical ranking functions approach is reduced substantially, and second, global ranks are obtained from local information. However, in many practical applications, even the local rankings are only available in parts, or not exactly in the format that is needed. In this paper, we apply inductive reasoning methods like System Z+ or c-representations, to fill up missing values in the local conditional tables. This allows the user to specify knowledge for such OCF-networks in its most appropriate and reliable form and leave the technical details to an inference engine.

1 Introduction

Uncertain and defeasible reasoning is often crucially based on appropriate semantical frameworks like, for example, probability theory that allow for a rich and meaningful representation of the problem domain under consideration while leaving enough semantical room for handling exceptions and nonmonotonic phenomena. One of these frameworks is provided by the theory of ordinal conditional functions (OCF) [1], also called ranking functions, that assign a degree of disbelief to any possible world. Ranking functions have become increasingly popular within the last decade, as they are essentially qualitative and more easily understandable than probabilities but share some nice features with probabilities. Most importantly, they provide proper interpretations for (meaningful, non-material) conditionals \( (B|A) \) – If A then plausibly B, encoding a plausible relationship between their antecedents A and consequents B.

However, the drawback of most semantical approaches is their high complexity as query answering and inference procedures have to take (basically) all models into account. To make local computations considering only a subset of all variables possible, graphical structures like Bayesian networks [2] have proved extremely useful. Usually, they come along with a causal interpretation considering the parents of a variable as its (common) causes. For OCFs, a similar type of networks has been proposed in [3,4]. In these approaches, analogous to Bayesian networks, OCFs are factorized according to the structure of a graph, and local ranking tables involving only a few nodes serve to build a global ranking function. Still, as in Bayesian networks, the local ranking tables need full information of how plausible a literal is given all configurations of the parents.
of the respective variable. In application scenarios, often only partial information is available here, typically, the user only knows the plausibility of a variable given each cause separately and cannot say much about cases when some configuration of causes is present. To fill up information, often external combination rules like naive Bayes [5] are applied which, however, do not take the semantical structure of the problem under consideration into account.

In this paper, we propose methods to combine partial ranking information in an intensional way to come up with full local ranking tables so that the OCF-network and hence the induced global ranking function can be completely specified. The basic idea is to apply inductive conditional reasoning mechanisms like System Z+ [6] and c-representations [7] locally to find appropriate (complete) rankings for the respective subgraph, that is, a node and its parents, and extract from this semantical information the missing values in the local table of the child node. Similar approaches have been presented for Bayesian networks by making use of the maximum entropy principle [8,9,10]. Indeed, the maximum entropy distribution is a probabilistic c-representation for the given knowledge base [7], and for the OCF framework, inferences based on c-representations have also proved to satisfy all major properties of nonmonotonic reasoning [11]. Therefore we make use of high quality semantical methods to exploit the given partial information in an optimal way. This paper is an extended version of the paper [12]; in particular, we elaborated on formal properties of OCF-networks in much more detail. The rest of this paper is organized as follows: After short preliminaries in Section 2 we recall ranking functions in Section 3. Then, in Section 4, we introduce the systems to be used for inductive conditional reasoning, namely System Z+ and c-representations. In Section 5 we elaborate on the concept of networks for ranking functions. We discuss why local ranks may not be available in the needed format for all vertices in the network and show how to solve this problem with the presented approaches in Section 6. Finally, we conclude in Section 6.

2 Preliminaries

Let \( \Sigma = \{V_1, \ldots, V_n\} \) be a set of propositional atoms and a literal a positive or negative atom representing variables in their positive resp. negated form; for a specific, nevertheless undetermined, outcome of \( V_i \), we write \( \dot{v}_i \in \{v_i, \bar{v}_i\} \).

The set of formulas \( \mathcal{L} \) over \( \Sigma \) joined with the symbols for tautology (\( \top \)) and contradiction (\( \perp \)), with the connectives \( \land \) (and), \( \lor \) (or) and \( \neg \) (not) shall be defined in the usual way. For \( A, B \in \mathcal{L} \), we usually omit the connective \( \land \) and write \( AB \) instead of \( A \land B \) as well as indicate negation by overlining, that is, \( \overline{A} \) means \( \neg A \).

Interpretations, or possible worlds, are also defined in the usual way; the set of all possible worlds is denoted by \( \Omega \). We often use the 1-1 association between worlds and complete conjunctions, that is, conjunctions of literals where every variable \( V_i \in \Sigma \) appears exactly once. A model \( \omega \) of a propositional formula \( A \in \mathcal{L} \) is a possible world that satisfies \( A \), written as \( \omega \models A \). The set of all models \( \omega \models A \) is denoted by \( \text{Mod}(A) \).

For formulas \( A, B \in \mathcal{L} \), \( A \) entails \( B \), written as \( A \models B \), iff \( \text{Mod}(A) \subseteq \text{Mod}(B) \), that is, if and only if for all \( \omega \in \Omega, \omega \models A \) implies \( \omega \models B \). For sets of formulas \( A \subseteq \mathcal{L} \) we have \( \text{Mod}(A) = \bigcap_{A \in A} \text{Mod}(A) \).
Fig. 1. Ranking model $\kappa$ of the penguin example $\mathcal{R} = \{(f|b), (\overline{f}|p), (b|p)\}$ given as worlds stacked by their plausibility (left) and in tabular form (right), and ranking models $\kappa_Z$ and $\kappa_R$ of the annotated penguin knowledge base in Example 1.

A conditional $(B \mid A)$ with $A, B \in \mathcal{L}$ encodes a defeasible rule “if $A$ then usually $B$” with the trivalent evaluation $[(B \mid A)]_\omega = true$ if and only if $\omega \models AB$ (verification), $[(B \mid A)]_\omega = false$ if and only if $\omega \models A\overline{B}$ (falsification) and (non-applicability) $[(B \mid A)]_\omega = undefined$ if and only if $\omega \models \overline{A}$ [13,11]. The language of all conditionals over $\mathcal{L}$ is denoted by $\mathcal{L}$. We denote by $\mathcal{R}$ a finite set of conditionals $\mathcal{R} = \{(B_1 \mid A_1), \ldots , (B_n \mid A_n)\} \subseteq (\mathcal{L} \mid \mathcal{L})$. A conditional $(B \mid A)$ is tolerated by $\mathcal{R}$ if and only if there is a world $\omega \in \Omega$ such that $\omega \models AB$ and $\omega \models A_i \Rightarrow B_i$ for every $1 \leq i \leq n$. $\mathcal{R}$ is consistent if and only if for every nonempty subset $\mathcal{R}' \subseteq \mathcal{R}$ there is a conditional $(B \mid A) \in \mathcal{R}'$ that is tolerated by $\mathcal{R}'$ [3]. We call such a consistent $\mathcal{R}$ a knowledge base and it shall represent the knowledge an agent uses as a base for reasoning.

3 Ranking Functions (OCF)

An ordinal conditional function (OCF, [1]), also called ranking function, is a function $\kappa : \Omega \rightarrow \mathbb{N}_0^\infty$ with $\kappa^{-1}(0) \neq \emptyset$ which maps each world $\omega \in \Omega$ to a degree of implausibility $\kappa(\omega)$, that is, if for two possible worlds $\omega, \omega' \in \Omega$ it holds that $\kappa(\omega) < \kappa(\omega')$ then $\omega'$ is believed to be less plausible than $\omega$. Ranks of formulas $A \in \mathcal{L}$ are calculated as $\kappa(A) = \min \{\kappa(\omega) \mid \omega \models A\}$. For conditionals $(B \mid A)$ a rank is defined via $\kappa(B \mid A) = \kappa(AB) - \kappa(A)$. A ranking function $\kappa$ is a (ranking) model of a conditional $(B \mid A)$, written $\kappa \models (B \mid A)$ if and only if $\kappa(AB) < \kappa(A\overline{B})$, that is, if and only if $AB$ is more plausible than $A\overline{B}$. Figure 1 is an example for a ranking model of the knowledge base for the well-known penguin-example, $\mathcal{R} = \{(f|b), (\overline{f}|p), (b|p)\}$, encoding the rules “birds usually fly”, “penguins usually do not fly” and “penguins usually are birds”. If $\kappa \models (B \mid A)$ we also say that $(B \mid A)$ is believed in $\kappa$.

Sometimes it is convenient to express how strongly a conditional is believed.

**Definition 1 (Firmness [1]).** A proposition $A$ is believed in an OCF $\kappa$ with firmness $m$, $m \in \mathbb{N}$, if and only if $\kappa(A) \geq m$. A conditional $(B \mid A)$ is believed with firmness $m$ ($\kappa \models (B \mid A)[m]$), if and only if $\kappa(B \mid A)[m] \geq m$.

Equivalently, $(B \mid A)$ is believed in $\kappa$ with firmness $m$ iff $\kappa(AB) + m \leq \kappa(A\overline{B})$. Moreover, $\kappa$ is a model of $(B \mid A)$ if the conditional is believed with firmness $m = 1$, that is, if $\kappa(AB) + 1 \leq \kappa(A\overline{B})$ which is equivalent to $\kappa(AB) < \kappa(A\overline{B})$. We illustrate the notion of firmness with the following example.
Example 1. Let $\mathcal{R} = \{ (f|b)[1], (\overline{f}|p)[2], (b|p)[10] \}$ be the penguin knowledge base with annotated conditionals. Two OCFs, $\kappa_Z$ and $\kappa_{\overline{Z}}$, that are models of $\mathcal{R}$ are shown in Fig. 1.

Note that $\kappa \models A[m]$ if and only if $\kappa \models (A|\top)[m]$ since $\kappa(\top) = 0$, so (plausible) formulas can be considered as a special case of conditionals. Hence, we focus on conditional knowledge bases in this paper, keeping in mind that such knowledge bases may also contain plausible propositions. Moreover, we presuppose $m \geq 1$ in this paper since $\kappa \models (B|A)[m]$ should imply in particular $\kappa \models (B|A)$. Nevertheless, the case $m = 0$ is interesting but requires further considerations as we might have $\kappa(AB) = \kappa(A\overline{B})$, or $\kappa \models (B|A)$. To keep the technical details as clear and simple as possible, we leave the case $m = 0$ for future work.

Definition 2 (Admissibility). A ranking function is admissible regarding a knowledge base of conditionals $\mathcal{R} = \{(B_1|A_1)[m_1], \ldots, (B_n|A_n)[m_n]\}$ annotated with firmness values (formally, $\kappa \models \mathcal{R}$) if and only if $mbox{bbox} \models (B_i|A_i)[m_i]$ for every $1 \leq i \leq n$.

For the networks to be considered in this paper, we need a notion of conditional independence regarding ranking functions [1], as it is necessary for Bayes networks.

Definition 3 (Conditional $\kappa$-independence [1]). Let $A,B,C$ be disjoint sets of variables. $A$ is (conditionally) $\kappa$-independent of $B$ given $C$, written $A \perp_\kappa B \mid C$ if and only if $\kappa(ab|c) = \kappa(a|c) + \kappa(b|c)$ for all complete conjunctions $a,b,c$ built over $A,B,C$, respectively.

$\kappa$-independence can be characterized equivalently as in the probabilistic case:

Lemma 1. For all disjoint sets of variables $A,B,C$, $A$ is $\kappa$-independent of $B$ given $C$ ($A \perp_\kappa B \mid C$) if and only if $\kappa(ab|c) = \kappa(a|c) + \kappa(b|c)$ for all complete conjunctions $a,b,c$ over $A,B,C$, respectively.

Proof. $A \perp_\kappa B \mid C$ means $\kappa(ab|c) = \kappa(a|c) + \kappa(b|c)$, therefore $\kappa(abc) - \kappa(c) = \kappa(a|c) + \kappa(bc) - \kappa(c)$. This is equivalent to $\kappa(abc) - \kappa(bc) = \kappa(a|c)$, hence $\kappa(ab|c) = \kappa(a|c)$, which was to be shown.

4 Inductive conditional reasoning

Let $\mathcal{R} = \{(B_1|A_1)[m_1], \ldots, (B_n|A_n)[m_n]\}$ be an annotated knowledge base. Taking all admissible ranking functions into account yields quite a weak inference from $\mathcal{R}$, because in this case, the set of inferences that could be drawn from $\mathcal{R}$ is the intersection of all $\mathcal{R}$-admissible OCFs, that is, the sceptical inference relation regarding $\mathcal{R}$. A popular approach to obtain more informative inferences from $\mathcal{R}$ is realised by selecting a “best” ranking model of $\mathcal{R}$ to be used for further inferences. In the following, we recall two approaches to obtain such a “best” ranking function for inductive model-based inference.
4.1 System Z+

A well known approach to compute a ranking function given an annotated knowledge base $\mathcal{R} = \{(B_1|A_1)|m_1\}, \ldots, (B_n|A_n)|m_n\}$ is System Z+ [6] which is a generalization of System Z [14]. Here, the tolerance condition (cf. Section 2) is extended to firmness annotated conditionals such that $\mathcal{R}$ tolerates $(D|C)[m]$ if the knowledge base $\mathcal{R}^* = \{(B|A)|(B|A)[m] \in \mathcal{R}\}$ tolerates $(D|C)$. We start by selecting the set of conditionals $\mathcal{R}_0 \subseteq \mathcal{R}$ which are tolerated by the whole knowledge base, so $\mathcal{R}_0$ consists of all conditionals $(B|A)[m]$ with the property that there is a world $\omega$ such that $\omega \models AB$ and $\omega \models (A_i \Rightarrow B_i)$ for each $(B_i|A_i)[m] \in \mathcal{R}$. These conditionals get a Z-value identical to their firmness, that is, $Z(B_i|A_i) = m_i$ for all $(B_i|A_i) \in \mathcal{R}_0$. We initialize $\mathcal{R} \mathcal{Z}$ to $\mathcal{R} \mathcal{Z} = \mathcal{R}_0$. In the iteration step we select the set of worlds $\Omega_{\mathcal{R} \mathcal{Z}}$ which solely falsify conditionals in $\mathcal{R} \mathcal{Z}$ and verify at least one conditional not in $\mathcal{R} \mathcal{Z}$. Each of these worlds is assigned a temporal $\kappa^*_\mathcal{Z}$-rank calculated as $\kappa^*_\mathcal{Z}(\omega) = \max_{(B_i|A_i) \in \mathcal{R} \mathcal{Z}} \{Z(B_i|A_i)|\omega \models A_i \overline{B_i}\} + 1$.

From $\Omega_{\mathcal{R} \mathcal{Z}}$ we take a world $\omega^*$ with the smallest $\kappa^*_\mathcal{Z}$-value, that is, a world $\omega^* \in \Omega_{\mathcal{R} \mathcal{Z}}$ such that $\kappa^*_\mathcal{Z}(\omega^*) = \min_{\omega \in \Omega_{\mathcal{R} \mathcal{Z}}} \{\kappa^*_\mathcal{Z}(\omega)\}$. By construction, for each $\omega \in \Omega_{\mathcal{R} \mathcal{Z}}$ there is at least one conditional $(B_i|A_i)[m_i] \in \mathcal{R} \setminus \mathcal{R} \mathcal{Z}$ that is verified by $\omega$. To each of these conditionals we assign the Z-value $Z(B_i|A_i) = \kappa^*_\mathcal{Z}(\omega^*) + m_i$ and add $(B_i|A_i)$ to $\mathcal{R} \mathcal{Z}$, obtaining a new set $\mathcal{R} \mathcal{Z}$ which we start the iteration again until $\mathcal{R} \mathcal{Z} = \mathcal{R}$. For further details and theoretical background, confer [6].

There is a world $\omega$ with $\omega \models AB$ for every conditional $(B|A)[m] \in \mathcal{R}$, if the starting knowledge base $\mathcal{R}$ is consistent. This world either does not falsify any conditional $(B_i|A_i)[m] \in \mathcal{R}$, then $(B|A)$ is an element of $\mathcal{R}_0$, or there is a conditional $(D|C)[n] \in \mathcal{R}$ with $\omega \models C \overline{D}$, but then, $\omega$ is chosen as a world in $\Omega_{\mathcal{R} \mathcal{Z}}$ at a time after $(D|C)$ was added to $\mathcal{R} \mathcal{Z}$. By this we get an associated Z-value $Z(B_i|A_i)$ for all the conditionals in $\mathcal{R}$ and from these values we obtain a ranking function $\kappa_\mathcal{Z}$ defined as

$$\kappa_\mathcal{Z}(\omega) = \begin{cases} 0 & \text{iff } \omega \text{ does not falsify any } (B_i|A_i) \\ \max_{\omega \models (A_i \overline{B_i})} \{Z(B_i|A_i)\} & \text{otherwise.} \end{cases}$$

Note that, differently from the original approach [6], instead of setting a world’s rank to $\kappa_\mathcal{Z}(\omega) = \max_{\omega \models (A_i \overline{B_i})} \{Z(B_i|A_i)\} + 1$, we set this rank to the value of $\kappa_\mathcal{Z}(\omega) = \max_{\omega \models (A_i \overline{B_i})} \{Z(B_i|A_i)\}$, for the admissibility of $\kappa$ wrt. $(B|A)$ ($\kappa \models (B|A)[m]$), we require $\kappa(AB) + m \leq \kappa(AB)$ and not, like in [6], $\kappa(AB) + m < \kappa(AB)$, but presuppose $m > 0$.

Example 2 (System Z+ penguins). We use the knowledge base from Example 1 to illustrate how this framework works, so let $\mathcal{R} = \{(f|b)[1], (\overline{f}|p)[2], (b|p)[10]\}$. For the first step, we get $\mathcal{R}_0 = \{(f|b)\} = \mathcal{R} \mathcal{Z}$, so $Z(f|b) = 1$. We obtain $\Omega_{\mathcal{R} \mathcal{Z}} = pb\overline{f}$ and $\kappa*\mathcal{Z} = 2$. We have $pb\overline{f} \models pb\overline{b}$ as well as $pb\overline{f} \models bp$ and we get $Z(\overline{f}|p) = 2 + 2 = 4$ and $Z(b|p) = 2 + 10 = 12$. The resulting ranking function $\kappa_\mathcal{Z}$ is shown in Figure 1.
4.2 c-representations

The framework of c-representation [11] generates ranking functions $\kappa^c_R$ for knowledge bases $R$ that are $R$-admissible and are based on the conditionals in the knowledge base and their structure, solely. In this section, we rejudge this approach.

**Definition 4 (c-representation [11]).** A c-representation of a knowledge base defined as $R = \{(B_1|A_1)[m_1], \ldots, (B_n|A_n)[m_n]\}$ is an OCF of the form

$$
\kappa^c_R(\omega) = \sum_{i=1}^{n} \kappa^c_i, \\
\kappa^c_i \in \mathbb{N}_0
$$

where the values $\kappa^c_i$ are penalty points for falsifying conditionals and have to be chosen to make $\kappa^c_R$ $R$-admissible, that is for all $1 \leq i \leq n$ it holds that $\kappa^c_R \models (B_i|A_i)[m_i]$. This is the case if and only if [7], (cf. Definition 1):

$$
\kappa^c_i \geq m_i + \min_{\omega = A_iB_i} \left\{ \sum_{i \neq j} \kappa^c_j \right\} - \min_{\omega = A_iB_i} \left\{ \sum_{i \neq j} \kappa^c_j \right\}
$$

(2)

A minimal c-representation is obtained by choosing $\kappa^c_i$ minimally for all $1 \leq i \leq n$. (Note that there may be several different minimal c-representations for a given $R$.)

**Example 3 (c-represented penguins).** We use the knowledge base from Example 1 to illustrate how this framework works, so let $R = \{(f|b)[1], (j|p)[2], (b|p)[10]\}$. For the $\kappa^c_i$ values of a c-representation we get, according to inequation (2), $\kappa_1^c \geq 1 + \min\{\kappa_2^c, 0\} - \min\{0\} = 1$, $\kappa_2^c \geq 2 + \min\{\kappa_1^c, \kappa_3^c\} - \min\{0\} = 2 + \min\{\kappa_1^c, \kappa_3^c\}$ and $\kappa_3^c \geq 10 + \min\{\kappa_1^c, \kappa_2^c\} - \min\{0\} = 10 + \min\{\kappa_1^c, \kappa_2^c\}$. This leads to a minimal c-representation with $\kappa_1^c = 1$, $\kappa_2^c = 3$, $\kappa_3^c = 11$ and the OCF $\kappa^c_R$ shown in Fig. 1.

5 OCF-Networks

In this section, we elaborate on the concept of networks for OCFs. First approaches that make crucial use of the idea of causality have been presented in [3,4]. However, like in Bayesian networks, causal interpretations are not mandatory for such networks although they support appropriate modellings of the problem domain. More importantly, it is the idea of conditional independence that provides the basis for factorising OCFs, i.e., for local representations of global ranking functions. So, we prefer to develop the approach of OCF-networks in full analogy to Bayesian networks (as far as possible), making assumptions underlying the works [3,4] explicit.

Let $\Gamma = (\mathcal{V}, \mathcal{E})$ be a directed, acyclic graph (DAG) with a set of vertices $\mathcal{V} = \{V_1, \ldots, V_n\}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We define the parents of a vertex $V$, $pa(V)$, as the direct predecessors of $V$ (i.e., $pa(V) = \{V'|(V', V) \in \mathcal{E}\}$) and the descendants of $V$, $desc(V)$, as the set of vertices $V'$ for which a path from $V$ to $V'$ exists in $\mathcal{E}$. The set of non-descendants of $V$ is the set of all vertices that are neither the parents nor the descendants of $V$, nor $V$ itself, so $nd(V) = \mathcal{V} \setminus (desc(V) \cup \{V\} \cup pa(V))$.

To connect a DAG with ranking information we define an OCF-Network as follows:
Definition 5 (OCF-Network). A DAG \( \Gamma = (\Sigma, \mathcal{E}) \) over a set of propositional atoms \( \Sigma \) is an OCF-network if each vertex \( V \in \Sigma \) is annotated with a table of local rankings \( \kappa_V(V|\text{pa}(V)) \) with (local) ranking values specified for every configuration of \( V \) and \( \text{pa}(V) \). According to the definition of ranking functions the local rankings must be normalised, i.e.,

\[
\min_{\dot{v}} \{ \kappa(\dot{v}|\text{pa}(V)) \} = 0 \quad \text{for every configuration of } \text{pa}(V),
\]

and for the local rankings to provide a global picture, we also need

\[
\min_\omega \sum \kappa_V(V(\omega)|\text{pa}(V)(\omega)) = 0,
\]

to be fulfilled where \( V(\omega) \) resp. \( \text{pa}(V)(\omega) \) indicates the outcome \( v \in \text{dom}(V) \) with \( \omega \models v \) resp. the configuration \( p \) of the variables in \( \text{pa}(V) \) with \( \omega \models p \).

The local ranking information in \( \Gamma \) can be used to define a global ranking function \( \kappa \) over \( \Sigma \) by applying the idea of stratification [3]: A ranking function \( \kappa \) is stratified relative to an OCF-network \( \Gamma \) if and only if

\[
\kappa(\omega) = \sum \kappa_V(V(\omega)|\text{pa}(V)(\omega)),
\]

for every world \( \omega \). With this stratification, given the tables of local rankings, we can generate a stratified OCF by formula (5). Condition (4) ensures that \( \kappa \) is indeed an OCF.

Example 4. As an illustration we use the penguin example already presented in Example 3 with a graph set up according to [3] and local conditional ranking values calculated as conditional ranks from the ranking function given in Example 3 shown in Figure 2, i.e., \( \kappa_P(B|P) = \kappa(B|P) \), \( \kappa_F(F|PB) = \kappa(F|PB) \), \( \kappa_P(P) = \kappa(P) \).

\[
\begin{array}{c|cccc}
B|P & \kappa_P(B|P) \\
\hline
b\|p & 0 \\
b\|\neg p & 1 \\
\neg b\|p & 0 \\
\neg b\|\neg p & 0 \\
\end{array}
\]

\[
\begin{array}{c|cccc}
F|PB & \kappa_F(F|PB) \\
\hline
f\|b\|p & 1 \\
\neg f\|b\|p & 0 \\
f\|\neg b\|p & 0 \\
\neg f\|\neg b\|p & 0 \\
\end{array}
\]

\[
\begin{array}{c|c}
P|\kappa_P(P) \\
\hline
p & 1 \\
\neg p & 0 \\
\end{array}
\]

Fig. 2. Network of the penguin-example.

Conversely, given a DAG \( \Gamma \) with vertices \( \Sigma \) and an OCF \( \kappa \) over \( \Sigma \) such that each vertex \( V \in \Sigma \) is \( \kappa \)-independent of its non-descendants given its parents, we obtain a stratification of \( \kappa \) relative to \( \Gamma \). This is stated in the following proposition.
Proposition 1. Let $\Sigma$ be a propositional alphabet and $\Gamma = (\Sigma, E)$ be a DAG. Let $\Sigma = \{V_1, \ldots, V_n\}$ be enumerated such that for each $V_i \in \Sigma$ we have $pa(V_i) \subseteq \{V_1, \ldots, V_{i-1}\}$. Let $\kappa$ be an OCF over $\Sigma$ such that $V \perp_{\kappa} nd(V) \mid pa(V)$ for all $V \in \Sigma$. Then it holds that

$$\kappa(V_1, \ldots, V_n) = \sum_{i=1}^{n} \kappa(V_i \mid pa(V_i)).$$

(6)

Proof. Let $\Sigma, \Gamma, \kappa$ be as presupposed in the lemma above. Then

$$\kappa(V_1, \ldots, V_n) = \kappa(V_1, \ldots, V_n) - \kappa(V_1, \ldots, V_{n-1}) + \kappa(V_1, \ldots, V_{n-1})$$

$$- \kappa(V_1, \ldots, V_{n-2}) + \kappa(V_1, \ldots, V_{n-2}) - \ldots - \kappa(V_1) + \kappa(V_1)$$

$$= \kappa(V_n \mid V_1, \ldots, V_{n-1}) + \kappa(V_{n-1} \mid V_1, \ldots, V_{n-2}) + \ldots + \kappa(V_1)$$

$$= \kappa(V_1) + \sum_{i=2}^{n} \kappa(V_i \mid V_1, \ldots, V_{i-1}).$$

By presupposition, $pa(V_i) \subseteq \{V_1, \ldots, V_{i-1}\}$ and $V_i \perp_{\kappa} nd(V_i) \mid pa(V_i)$ for all $V_i$. With lemma 1 we obtain the equality $\kappa(V_i \mid V_1, \ldots, V_{i-1}) = \kappa(V_i \mid pa(V_i))$. Therefore for the joint ranking function we have

$$\kappa(V_1, \ldots, V_n) = \sum_{i=1}^{n} \kappa(V_i \mid pa(V_i))$$

for every $1 \leq i \leq n$

which was to be shown.

Hence, a ranking function that implements the conditional independence assumptions of a network $\Gamma$ can be stratified relative to $\Gamma$. Whether a stratified ranking function is admissible with respect to the ranking tables is settled by the next theorem.

Theorem 1. Let $W$ be a variable in $\Sigma$ with a fixed value $\hat{w}$ of $W$, let $\hat{p}_w$ be a fixed configuration of the variables in $pa(W)$. For a ranking function $\kappa$ stratified according to equation (5) the conditional ranking values $\kappa(\hat{w} \mid \hat{p}_w)$ are identical to the local ranking values $\kappa_W(\hat{w} \mid \hat{p}_w)$.

Proof. According to the definitions of OCF and conditional ranking values in Section 3 we have

$$\kappa(\hat{w} \mid \hat{p}_w) = \kappa(\hat{w} \hat{p}_w) - \kappa(\hat{p}_w)$$

$$= \min_{\omega = \hat{w} \hat{p}_w} \{\kappa(\omega)\} - \min_{\omega = \hat{p}_w} \{\kappa(\omega)\}.$$

With the stratification of equation (5), this rewrites to

$$\kappa(\hat{w} \mid \hat{p}_w) = \min_{\omega = \hat{w} \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega) \mid pa(V)(\omega)) \right\} - \min_{\omega = \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega) \mid pa(V)(\omega)) \right\}.$$

The value of $\kappa_W(\hat{w} \mid \hat{p}_w)$ is fixed in every sum of the first minterm and hence can be extracted. For the first minterms we obtain with the normalisation condition (3) that
for every vertex $C$ in the set of children of $W$ there is a configuration $\hat{c}$ of $C$ such that $\kappa_C(\hat{c}|\hat{w}) = 0$, which holds iteratively for the children of $C$, so in the above formula, we have
$$\min_{\omega = \hat{w} | \hat{p}_w} \left\{ \sum_{V \in \text{desc}(W)} \kappa_V(V(\omega)|pa(V(\omega)) \right\} = 0.$$ Since the configuration of each vertex apart from $W$ and $W$’s parents is not fixed and the configuration of each variable is independent from the others, the actual minimum is achieved when a configuration as sketched above is chosen, hence the descendants of $W$ can be ignored for the first minterm. So we have
$$\kappa(\hat{w}|\hat{p}_w) = \min_{\omega = \hat{w} | \hat{p}_w} \left\{ \begin{array}{l}
\kappa_W(\hat{w}|\hat{p}_w) \\
+ \sum_{V \in \text{nd}(W) \cup \text{pa}(W)} \kappa_V(V(\omega)|pa(V(\omega))) \\
+ \sum_{V \in \text{desc}(W)} \kappa_V(V(\omega)|pa(V(\omega)))
\end{array} \right\}$$
$$= \kappa_W(\hat{w}|\hat{p}_w) + \min_{\omega = \hat{w} | \hat{p}_w} \left\{ \begin{array}{l}
\sum_{V \in \{\omega\} \cup \text{desc}(W)} \kappa_V(V(\omega)|pa(V(\omega)))
\end{array} \right\}$$
$$- \min_{\omega = \hat{w} | \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega)|pa(V(\omega))) \right\}.$$

The value of vertices $V \in \text{nd}(W) \cup \text{pa}(W)$ can be chosen independently of the value $W$, while $\hat{p}_w$ is fixed, hence the minimum of the sum over this values is constant for the term. It therefore can be extracted and will be called $\text{Const}$ in the following, so the equation can be written as
$$\kappa(\hat{w}|\hat{p}_w) = \kappa_W(\hat{w}|\hat{p}_w) + \text{Const} - \min_{\omega = \hat{w} | \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega)|pa(V(\omega))) \right\}.$$

We now look at the second minterm. Here, $W$ is not fixed so this variable can be chosen freely. Naturally, the minimum of this term is either the minimal sum with $W$ fixed to $w$ or to $\bar{w}$, so we can rewrite this to
$$\min_{\omega = \hat{w} | \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega)|pa(V(\omega))) \right\}$$
$$= \min \left\{ \begin{array}{l}
\min_{\omega = \hat{w} | \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega)|pa(V(\omega))) \right\} ^{\Sigma_1} \\
\min_{\omega = \bar{\hat{w}} | \hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega)|pa(V(\omega))) \right\} ^{\Sigma_2}
\end{array} \right\}.$$

We continue our elaborations on the case $\Sigma_1$, first. We partition the summation condition $V$ into in the set of $W$’s descendants, $W$ itself and all other vertices $(\text{nd}(W) \cup$
\(pa(W)\) and obtain for \(\Sigma_1\) the term

\[
\Sigma_1 = \min_{\omega = \hat{w}p_w} \left\{ \sum_{V \in (\text{nd}(W) \cup \text{pa}(W))} \kappa_V (V(\omega) | \text{pa}(V)(\omega)) + \sum_{V \in \text{desc}(W)} \kappa_V (V(\omega) | \text{pa}(V)(\omega)) \right\}.
\]

The minimum can be split, since the values of nodes in \((\text{nd}(W) \cup \text{pa}(W))\) can be chosen independently from those in \(\text{desc}(W)\). \(\kappa_W(w|p_w)\) can be extracted since the values of \(W\) and \(\hat{w}_w\) are fixed in the min. The sum over \(V \in (\text{nd}(W) \cup \text{pa}(W))\) is independent of the interpretation of \(W\), this minimum is the same constant \(\text{Const}\) as above. The minimum of the sum over \(V \in \text{desc}(W)\) is 0 as discussed above, so for this minimum we get \(\Sigma_1 = \text{Const} + \kappa_W(w|p_w)\). We extend this deliberations to \(\Sigma_2\) and obtain similarly \(\Sigma_2 = \text{Const} + \kappa_W(\bar{w}|\bar{p}_w)\). So finally we obtain the following for the second minterm.

\[
\begin{align*}
\kappa(\hat{w}|\hat{p}_w) &= \kappa_W(w|p_w) + \text{Const} - \text{Const}.
\end{align*}
\]

And we obtain \(\kappa(\hat{w}|\hat{p}_w) = \kappa_W(w|p_w)\) as proposed. \(\square\)

**Theorem 2.** Let \(\Gamma = \langle \Sigma, \mathcal{E} \rangle\) be an OCF-network with local ranking tables \(\kappa_V(V|pa(V))\) for every vertex \(V \in \Sigma\) and stratified OCF \(\kappa\) according to (5). Then the local directed Markov property holds, that is, \(V \perp_{\kappa} \text{nd}(V) \mid pa(V)\) for each node \(V\).

**Proof.** Let \(\Gamma, \kappa_V\) and \(\kappa\) be as described in the theorem. For each node \(V\), let \(\hat{n}_v\) be a configuration of the variables \(\text{nd}(V)\) and \(\hat{p}_v\) be a configuration of the variables \(\text{pa}(V)\). We consider a fixed but arbitrary variable \(W\) in \(\Sigma\). We will show that

\[
\kappa(\hat{w}\hat{n}_w|\hat{p}_w) = \kappa(\hat{w}|\hat{p}_w) + \kappa(\hat{n}_w|\hat{p}_w)
\]

for every configuration of \(\hat{w}\), \(\hat{n}_w\) and \(\hat{p}_w\), this establishes the local Markov property as claimed. From theorem 1 we obtain \(\kappa(\hat{w}|\hat{p}_w) = \kappa_W(w|p_w)\), so by the definition of conditional ranks, it is equivalent to show \(\kappa(\hat{w}\hat{n}_w\hat{p}_w) = \kappa_W(w|p_w) + \kappa(\hat{n}_w|\hat{p}_w)\). We consider the left hand side first:

\[
\kappa(\hat{w}\hat{n}_w\hat{p}_w) = \min_{\omega = \hat{w}\hat{n}_w\hat{p}_w} \left\{ \sum_{V} \kappa_V(V(\omega)|\text{pa}(V)(\omega)) \right\}
\]

\[
= \min_{\omega = \hat{w}\hat{n}_w\hat{p}_w} \left\{ \sum_{V \in \text{nd}(W) \cup \text{pa}(W)} \kappa_V(V(\omega)|\text{pa}(V)(\omega)) + \sum_{V \in \text{desc}(W)} \kappa_V(V(\omega)|\text{pa}(V)(\omega)) \right\}.
\]
The first sum here is fixed by the chosen configuration $\hat{n}_w, \hat{p}_w$, the min over the second sum is 0, as discussed in the proof of theorem 1:

$$\sum_{V \in \text{nd}(W) \cup \text{pa}(W)} \kappa_V(V|pa(V)(\omega)) =: \text{Const}(\hat{n}_w, \hat{p}_w),$$

$$\min_{\omega=\hat{n}_w \hat{p}_w} \left\{ \sum_{V \in \text{desc}(W)} \kappa_V(V|pa(V)(\omega)) \right\} = 0,$$

hence $\kappa(\hat{n}_w \hat{p}_w) = \kappa_W(\hat{w}|\hat{p}_w) + \text{Const}(\hat{n}_w, \hat{p}_w)$. For the right hand side, we obtain similarly

$$\kappa(\hat{n}_w \hat{p}_w) = \min_{\omega=\hat{n}_w \hat{p}_w} \left\{ \sum_V \kappa_V(V|pa(V)(\omega)) \right\}$$

$$= \min_{\omega=\hat{n}_w \hat{p}_w} \left\{ \kappa_W(W(\omega)|\hat{p}_w) + \sum_{V \in \text{nd}(W) \cup \text{pa}(W)} \kappa_V(V|pa(V)(\omega)) + \sum_{V \in \text{desc}(W)} \kappa_V(V|pa(V)(\omega)) \right\}$$

$$= \text{Const}(\hat{n}_w, \hat{p}_w) + \min_{\omega=\hat{n}_w \hat{p}_w} \left\{ \kappa_W(W(\omega)|\hat{p}_w) \right\} = \text{Const}(\hat{n}_w, \hat{p}_w),$$

since $\min_{\omega=\hat{n}_w \hat{p}_w} \left\{ \sum_{V \in \text{desc}(W)} \kappa_V(V|pa(V)(\omega)) \right\} = 0$, independently of $W$ having the value $w$ or $\overline{w}$, and one of $\kappa_W(w|\hat{p}_w), \kappa_W(\overline{w}|\hat{p}_w)$ again must be 0.

Hence $\kappa(\hat{n}_w \hat{p}_w) = \kappa_W(\hat{w}|\hat{p}_w) + \text{Const}(\hat{n}_w, \hat{p}_w) = \kappa_W(\hat{w}|\hat{p}_w) + \kappa(\hat{n}_w \hat{p}_w)$, and this completes the proof. \(\square\)

6 Filling in gaps by intensional combination

OCF-networks and stratifications are most valuable concepts for practical applications of the ranking framework as they help to cut down the complexity of full semantical information. However, one often has also to struggle with the problem of incomplete information, that is, only some (conditional) relationships between variables can be expressed with sufficient reliability. Typically, experts are quite certain about stating relationships between variables and each of its causes, or between variables and special configurations of its parents. In these cases, we first have to fill in missing values in the local ranking tables by somehow exploiting the partial explicit information, before we can apply the OCF-networks approach.

Hence we aim at calculating missing lines in the local table of ranking values $\kappa_V(V|pa(V))$ exploiting the available knowledge as well as possible and use inductive,
intensional inference mechanisms like c-representations and System Z\(^+\) on local knowledge. From these local ranking functions, we can easily read the missing tabular values for \(V\) and fill up the complete local tables. More precisely, the procedure for filling in missing values in the ranking tables is as follows:

Let a DAG \(\Gamma\) over \(\Sigma\) be given, and for each \(V \in \Sigma\), let \(\mathcal{R}_V\) be a local conditional knowledge base containing statements of the form \((v|A)[m]\) where \(A\) is a formula involving only the parents of \(V\). For example, \(\mathcal{R}_V\) might have the form \(\mathcal{R}_V = \{(\hat{v}|\hat{A}))[m]|v_i \in pa(V)\}\.

In cases where \(\mathcal{R}_V\) is not a complete conditional ranking table, do the following:

1. Consider \(\mathcal{R}_V\) as a knowledge base over \(\Sigma' = \{V\} \cup pa(V)\).
2. Compute an OCF \(\kappa_V\) over \(\Sigma'\) from \(\mathcal{R}_V\) by using an inductive conditional reasoning method, like System Z\(^+\) or c-representations (cf. Section 4).
3. Compute from \(\kappa_V\) complete ranking tables \(\kappa_V(V|pa(V))\) for every configuration of \(V\) and \(pa(V)\).

Example 5. As an illustration, we modify an example from [3,4] that extends the example given in [12] by considering more complex information. A car starts \((S = s)\) if the battery is charged \((B = b)\) and the fuel tank is full \((F = f)\). If either the battery is discharged \((B = \bar{b})\) or the fuel tank is empty \((F = \bar{f})\), the car does not start \((S = \bar{s})\); additionally, if, for some reason, the headlights have been left switched on overnight \((H = h)\), the battery is discharged. We assume to know that it is very implausible to have left the headlights switched on \((\kappa_H(h) = 15)\) and usually the tank is not empty \((\kappa_F(\bar{f}) = 10)\). We also know that if the headlights have been switched on overnight, the battery is plausibly discharged \((\kappa_B(b|h) = 4)\) but if the headlights have been switched off, the battery usually is charged \((\kappa_B(b|\bar{h}) = 8)\). Unfortunately, we are unaware of many ranking values at vertex \(S\), in fact we just know that it is highly implausible that cars with an discharged battery and a full fueltank or cars with an empty fueltank but charged battery can be started \((\kappa_S(s|\bar{b}f) = 11, \kappa_S(s|\bar{b}\bar{f}) = 13)\). However, we know that it is highly implausible for a car with an empty battery to start, that is \(\kappa_S(s|\bar{b}) = 12\), and even less plausible that a car without any fuel will start, that is \(\kappa_S(s|\bar{f}) = 15\). On the other hand a car with a loaded battery usually starts \(\kappa_S(\bar{s}|b) = 2\) and a car with a full fueltank should start, too, \(\kappa_S(\bar{s}|f) = 1\). The OCF-network to this situation is shown in Figure 3, the local knowledge base from the joint information regarding \(S\) is

\[
\mathcal{R}_S^* = \{r_1 = (s|b)[2], r_2 = (\bar{s}|\bar{b})[12], r_3 = (s|f)[1], r_4 = (\bar{s}|\bar{f})[15], r_5 = (s|\bar{f})[13], r_6 = (\bar{s}|\bar{f})[11]\}.
\]

In this situation, we search for a local ranking function on \(S, B, F\) from which we can obtain the ranks of the vertices given all its parents. This can be achieved by using inductive conditional reasoning, i.e., by applying the methods presented in section 4. Note that, in constrast to [12], the local available knowledge at vertex \(S\) contains a more complex mixture of information regarding \(S\). Some information is specialised enough to provide aneries for the local table, but still the table has to be vompelted by using the the more general information for inductive reasoning First, we apply System Z\(^+\). We compute the partition \(\mathcal{R}_0\) of tolerated conditionals for this approach and find that \(\mathcal{R}_0\) consists of the conditionals \(r_1, r_2, r_3\) and \(r_4\).
Therefore we can assign to each conditional in $\mathcal{R}_0$ the Z-value given as firmness in $\mathcal{R}_S$ and set $Z(r_1) = 2$, $Z(r_2) = 12$, $Z(r_3) = 1$ and $Z(r_4) = 15$. In the next steps of the algorithm we add, with a minimal world $b\overline{f}\overline{s}$, $r_5$ to $\mathcal{R}_0$ giving it the Z-value of $Z(r_5) = \kappa_5^Z(b\overline{f}\overline{s}) + 13 = 16$, followed by $r_6$ with a minimal world $bfs$ and a Z-value of $Z(r_6) = \kappa_6^Z(bfs) + 11 = 13$.

We then set up a table indicating verification/falsification of the conditionals in $\mathcal{R}_S$ for each configuration of the local variables $B$, $F$ and $S$, and associate with them the ranks according to formula (1). So we obtain the local ranking function $\kappa_S^Z(BFS)$ shown in Table 1. This table also proves useful to set up the inequalities needed to calculate a c-representation of the knowledge base according to inequation (2). Here we obtain $\kappa_1^c \geq 2 + \min\{0, \kappa_4^c + \kappa_5^c\} - \min\{0, \kappa_3^c\} = 2$, $\kappa_2^c \geq 12 + \min\{0, \kappa_7^c\} - \min\{\kappa_6^c, \kappa_4^c\}$, $\kappa_3^c \geq 1 + \min\{0, \kappa_2^c + \kappa_6^c\} - \min\{0, \kappa_1^c\} = 1$, $\kappa_4^c \geq 15 + \min\{0, \kappa_7^c\} - \min\{\kappa_5^c, \kappa_2^c\}$, $\kappa_5^c \geq 13 + \min\{\kappa_1^c\} - \min\{\kappa_4^c\}$ and $\kappa_6^c \geq 11 + \min\{\kappa_3^c\} - \min\{\kappa_2^c\}$. We choose one and set $\kappa_5^c = 0$ and $\kappa_6^c = 0$ and by this we get $\kappa_2^c = 12$ and $\kappa_4^c = 15$. By this we get a c-representation $\kappa_S^c(BFS)$ also shown Table 1.

**Table 1.** Verification/falsification behaviour of configurations over the variables \{B, F, S\} given the local car start knowledge base from Figure 3.

| BFS verifies | falsifies $\kappa_S^Z(BFS)$ | $\kappa_S^Z(BF)$ | $\kappa_S^Z(S|BF)$ | $\kappa_S^Z(BFS)$ | $\kappa_S^Z(S|BF)$ |
|--------------|----------------------------|------------------|-------------------|------------------|-------------------|
| $bfs$        | $r_1, r_3$                 | 0                | 0                 | 0                | 0                 |
| $b\overline{f}\overline{s}$ | $r_1, r_3$              | 2                | 0                 | 3                | 0                 |
| $b\overline{f}\overline{s}$ | $r_4, r_5$              | 16               | 2                 | 15               | 2                 |
| $b\overline{f}\overline{s}$ | $r_3$                    | 2                | 2                 | 2                | 2                 |
| $\overline{f}s$ | $r_3$                   | 13               | 1                 | 12               | 1                 |
| $\overline{f}s$ | $r_2, r_6$              | 1                 | 0                 | 1                | 1                 |
| $\overline{f}s$ | $r_2, r_4$              | 15               | 0                 | 15               | 0                 |
| $\overline{f}s$ | $r_2, r_4$              | 0                | 0                 | 0                | 0                 |

Fig. 3. Problem description of the car Example 5.
Using this c-representation we can distinguish here between the configurations $b\, f\, s$ and $b\, s\, f$ whereas this is not the case is we use the values calculated with System $Z^+$. With the determined values, we complete the local conditional ranking table $(S\mid BF)$ by calculating $\kappa_S(s\mid b\, f) = \kappa_S(b\, f\, s) - \kappa_S(b\, s)$ for either one of the approaches shown in Table 1, too, and complete the graph from Fig. 3 to the OCF-network shown in Fig. 4.

**Conclusion**

In this paper we investigated Bayesian-style networks annotated with Spohn’s ordinal conditional functions in place of probabilities, where values at some of the network’s vertices are not specified. We demonstrated that well-known inductive reasoning approaches like System $Z^+$ as well as c-representations are capable of filling up the missing values with respect to the semantical structure of the problem. This application-oriented result helps us accomplishing the goal to allow the user of a system based on an OCF-network to specify her knowledge in an appropriate way and still rely on network techniques, leaving the technical details regarding local tables to the mentioned inference mechanisms. Previously to this, we examined whether formal properties of Bayesian networks are valid for OCF-networks, as well. More precisely, we proved that the local Markov property is valid for OCF-networks and showed that the global rankings of the stratified OCF coincides with the local rankings in the tables. In our continuing work, we explore these ideas for efficient implementations of OCF-based knowledge representation.

**Acknowledgment:** This work was supported by Grant KI 1413/5 – 1 to Prof. Dr. Gabriele Kern-Isberner from the Deutsche Forschungsgemeinschaft (DFG) as part of the priority program “New Frameworks of Rationality” (SPP 1516). Christian Eichhorn is supported by this grant.
References

An Overview of Algorithmic Approaches to Compute Optimum Entropy Distributions in the Expert System Shell MECore

Nico Potyka, David Marenke, Engelbert Mittermeier
Dept. of Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany

Abstract. The expert system shell MECore provides a series of knowledge management operations to define probabilistic knowledge bases and to reason under uncertainty. We report on our ongoing work regarding further development of MECore’s algorithms to compute optimum entropy distributions. We provide some intuition for these methods and point out their benefits as well as possible limitations and pitfalls.

1 Introduction

When designing expert systems, classical logics fail to meet the demands of reality, as they cannot deal with uncertain information. Probabilistic logics [22,1,25] are a natural extension that provide a framework to deal with uncertainty. Numerous approaches have been developed to deal with the accompanying computational complexity [16,12,28] or to deal with the combination of statistical data and expert knowledge [24]. We focus on probabilistic conditional logics [29,18,13] here. Experts can define intuitive rules of the form ‘if A then B with probability x’, which are formalized by conditionals (B|A)[x]. A and B can be arbitrary logical formulas and x is a probability that can express an expert’s degree of belief or can be obtained by statistical means.

The expert system shell MECore [8] allows the knowledge engineer to enter conditional knowledge bases and to accomplish probabilistic reasoning under the principle of optimum entropy [25,13]. A comprehensive case study can be found in [2], where a knowledge base for analysing brain tumor data is designed by both expert knowledge and statistical data. One of the most important operations for MECore is to adapt the current epistemic state, which is basically represented by a probability distribution, with respect to new knowledge. This is accomplished by computing I-projections, i.e., by computing a new epistemic state that satisfies the new knowledge and is closest to the old state. The ‘distance’ between two such epistemic states is measured by the relative entropy [4].

We report on our ongoing work on improving the efficiency of computing I-projections in the context of conditional knowledge bases. In Section 2, we recall the probabilistic logical framework and the basics of the principle of optimum entropy as they are needed in the following. In Section 3, we provide some intuition for MECore’s original I-projection algorithm (IIP) and a novel implementation.
based on \( L-BFGS \) [17]. In Section 4, we summarize some approaches to speed up the computation by taking the structure of conditional knowledge bases into account [13,7]. We also provide some experimental results to illustrate the benefit and shortcomings of different approaches.

2 Basics

2.1 Probabilistic Conditional Logic and Optimum Entropy

We consider a propositional conditional language \( \mathcal{L} \) built up over a finite set of propositional variables \( \Sigma \) using logical connectives like conjunction and disjunction in the usual way. For formulas \( \psi, \phi \in \mathcal{L} \) we abbreviate negation \( \neg \psi \) by an overbar \( \bar{\psi} \) and conjunction \( \psi \land \phi \) by juxtaposition \( \psi \phi \). A possible world is a classical logical interpretation \( \omega : \Sigma \rightarrow \{0,1\} \) assigning a truth value to each propositional variable. We also allow multi-valued variables that are associated with a domain and an interpretation assigns to each such variable a value from the corresponding domain. Let \( \Omega \) denote the set of all possible worlds. An atom \( a \) is satisfied by \( \omega \) if \( \omega(a) = 1 \). More generally, for a multi-valued variable \( X, x \in \text{domain}(X) \), the multi-valued atom \( X = x \) is satisfied by \( \omega \) if \( \omega(X) = x \).

The definition is extended to complex formulas in the usual way. The classical models \( \text{Mod}(\psi) \) of a formula \( \psi \) are the possible worlds \( \omega \in \Omega \) that satisfy \( \psi \).

We built up a probabilistic conditional language \( (\mathcal{L}|\mathcal{L}) := \{(\psi|\phi)[x] | \psi, \phi \in \mathcal{L}, x \in [0,1]\} \) over \( \mathcal{L} \). Intuitively a conditional \( (\psi|\phi)[x] \) expresses that our belief in \( \psi \) given that \( \phi \) holds is \( x \). Probabilistic interpretations \( \mathcal{P} : \Omega \rightarrow [0,1] \) are probability distributions over possible worlds. For a formula \( \psi \in \mathcal{L} \) we define \( \mathcal{P}(\psi) = \sum_{\omega \in \text{Mod}(\psi)} \mathcal{P}(\omega) \). A probability distribution \( \mathcal{P} \) satisfies a conditional \( (\psi|\phi)[x] \) iff \( \mathcal{P}(\psi|\phi) = x \cdot \mathcal{P}(\phi) \). If \( \mathcal{P}(\phi) > 0 \) it corresponds to the definition of conditional probability \( \mathcal{P}(\psi | \phi) = \frac{\mathcal{P}(\psi \phi)}{\mathcal{P}(\phi)} = x \).

A conditional knowledge base \( \mathcal{R} \subset (\mathcal{L}|\mathcal{L}) \) is a set of conditionals. A probability distribution \( \mathcal{P} \) satisfies \( \mathcal{R} \) iff it satisfies each \( c \in \mathcal{R} \). Let \( \text{Mod}(\mathcal{R}) \) denote the set of probabilistic models of \( \mathcal{R} \), i.e., the set of probability distributions over \( \Omega \) that satisfy \( \mathcal{R} \). If \( \text{Mod}(\mathcal{R}) \neq \emptyset \), \( \mathcal{R} \) is called consistent. One way to accomplish probabilistic reasoning with consistent knowledge bases is to select a best distribution \( \mathcal{P}^* \in \text{Mod}(\mathcal{R}) \). Then this distribution can be applied to compute probabilities for arbitrary formulas or conditionals.

An appropriate selection criterion is the principle of optimum entropy as it satisfies several common sense principles [25,13]. Among all distributions in \( \text{Mod}(\mathcal{R}) \) that satisfy \( \mathcal{R} \), one selects the unique one that maximizes the entropy \( H(\mathcal{P}) = -\sum_{\omega \in \Omega} \mathcal{P}(\omega) \log \mathcal{P}(\omega) \). More generally, given a prior distribution \( \mathcal{P}_p \), one selects the unique distribution in \( \text{Mod}(\mathcal{R}) \) that minimizes the relative entropy \( R(\mathcal{P}, \mathcal{P}_p) = \sum_{\omega \in \Omega} \mathcal{P}(\omega) \log \frac{\mathcal{P}(\omega)}{\mathcal{P}_p(\omega)} \) with respect to \( \mathcal{P}_p \). The unique solution \( \mathcal{P}^* \) is also called the I-projection of \( \mathcal{P}_p \) onto \( \text{Mod}(\mathcal{R}) \) [4] and is compactly written as \( \mathcal{P}^* = \mathcal{P} * \mathcal{R} \) [14] in the following. As maximizing entropy corresponds to minimizing relative entropy with respect to the uniform distribution \( \mathcal{P}_0 \), the maximum entropy solution is just a special I-projection.
As it is useful in the following, we identify conditionals with constraint functions restricting the possible models of a knowledge base. As explained before, \( \mathcal{P} \) satisfies a conditional \( c = (\psi|\phi)[x] \) iff \( \mathcal{P}(\psi\phi) = x \cdot \mathcal{P}(\phi) \). Subtracting \( x \cdot \mathcal{P}(\phi) \) from both sides of the equation yields

\[
0 = \mathcal{P}(\psi\phi) - x \cdot \mathcal{P}(\phi) = \mathcal{P}(\psi\phi) - x \cdot (\mathcal{P}(\psi\phi) + \mathcal{P}(\overline{\psi}\phi)) = (1 - x) \cdot \mathcal{P}(\psi\phi) - x \cdot \mathcal{P}(\overline{\psi}\phi)
\]

\[
= (1 - x) \cdot \left( \sum_{\omega \in \operatorname{Mod}(\psi\phi)} \mathcal{P}(\omega) \right) - x \cdot \left( \sum_{\omega \in \operatorname{Mod}(\overline{\psi}\phi)} \mathcal{P}(\omega) \right)
\]

\[
= \sum_{\omega \in \Omega} \mathcal{P}(\omega) \cdot (1_{\{\psi\phi\}}(\omega) \cdot (1 - x) - 1_{\{\overline{\psi}\phi\}}(\omega) \cdot x),
\]

where for a formula \( \phi \) the indicator function \( 1_{\{\phi\}}(\omega) \) maps to 1 iff \( \omega \) satisfies \( \phi \) and to 0 otherwise. Let \( f_c(\mathcal{P}) := \sum_{\omega \in \Omega} \mathcal{P}(\omega) \cdot (1_{\{\psi\phi\}}(\omega) \cdot (1 - x) - 1_{\{\overline{\psi}\phi\}}(\omega) \cdot x) \). Then \( \mathcal{P} \) is a model of \( c \) iff \( f_c(\mathcal{P}) = 0 \).

### 2.2 MECore

The workflow with MECore [8] can be described as follows. Initially the knowledge engineer defines a set of propositional variables \( \Sigma \). These variables can be multi-valued, for example in the medical case study in [2] a variable diagnosis is multi-valued, for example in the medical case study in [2] a variable diagnosis is multi-valued, for example in the medical case study in [2] a variable diagnosis is.

More strictly speaking, an epistemic state is described by a knowledge base \( \mathcal{R} \) reflecting the explicit knowledge and a probability distribution \( \mathcal{P} \) which satisfies the knowledge base \( \mathcal{R} \) and reflects the implicit knowledge. More formally, a belief base is a pair \((\mathcal{P}, \mathcal{R})\). The epistemic state corresponding to \((\mathcal{P}, \mathcal{R})\) is the I-projection of \( \mathcal{P} \) onto \( \operatorname{Mod}(\mathcal{R}) \) denoted by \( \mathcal{P} \cdot \mathcal{R} \). Hence, initially the belief base is \((\mathcal{P}_0, \emptyset)\) and the corresponding epistemic state is the uniform distribution \( \mathcal{P}_0 \).

MECore supports two belief change operations called revision and update [14]. They adapt the current epistemic state to new knowledge by computing special I-projections. The intuition is that revision deals with new knowledge in a static world, where the belief state may change but the explicit knowledge remains valid. Update deals with new knowledge in a dynamic world, where even the explicit knowledge may change. Therefore, revision in MECore presupposes that the new knowledge is consistent with the old explicit knowledge and extends this knowledge. Former explicit knowledge remains valid, even though the corresponding epistemic state usually changes. Update does not presuppose consistency with the old explicit knowledge and former explicit knowledge does not necessarily remain valid. Formally the revision operator \( \odot \) is defined by \((\mathcal{P}, \mathcal{R}) \odot \mathcal{R}^* := (\mathcal{P}, \mathcal{R} \cup \mathcal{R}^*)\). The update operator \( \ast \) is defined by \((\mathcal{P}, \mathcal{R}) \ast \mathcal{R}^* := (\mathcal{P} \cdot \mathcal{R}, \mathcal{R}^*)\). Note that the epistemic state after revision \((\mathcal{P}, \mathcal{R}) \odot \mathcal{R}^* \) is \( \mathcal{P} \cdot (\mathcal{R} \cup \mathcal{R}^*) \) and the epistemic state after update \((\mathcal{P}, \mathcal{R}) \ast \mathcal{R}^* \) is \( (\mathcal{P} \cdot \mathcal{R}) \ast \mathcal{R}^* \).
Also note that ∗ denotes the update operator as well as the I-projection operator. A more thorough discussion of the operators and their rationales can be found in [14].

A simple query for the brain tumor knowledge base from [2] is for example

\[ BT.query(diagnosis=meningeoma \mid (age = ge80) \text{ and } warningSymptoms), \]

which yields the belief in meningeoma to be the correct diagnosis given that the patient is older than 80 and warning symptoms can be observed. If further hypothetical knowledge has to be taken into account, what-if-analysis can be applied. A what-if-query is of the form \( \text{whatif}(R', \text{Query}) \) and yields the probability of the query \( \text{Query} \) taking the knowledge given by the conditional knowledge base \( R' \) into account. It corresponds to temporarily revising the current epistemic state with \( R' \) and answering the query.

2.3 Test sets

To compare performance of different approaches, in the following we consider three test sets shown in in Table 1. The table shows the number of (multi-valued) variables \( |\Sigma| \), the number of worlds \( |\Omega| \) and the number of conditionals \( |R| \).

| Name       | \( |\Sigma| \) | \( |\Omega| \)       | \( |R| \) |
|------------|--------------|----------------------|--------|
| TV         | 15           | 1.679.616            | 29     |
| CarDiagnosis | 18          | 6.718.464            | 38     |
| BrainTumor | 9            | 118.800              | 107    |

Table 1. Test sets.

TV describes technical features of TVs and fictitious relationships between them like \((size = big)|\text{and }display = lcd\)\(0.7\). CarDiagnosis reflects expert knowledge for fault detection for cars\(^1\). Finally, BrainTumor is the medical knowledge base described in [2] combining expert and statistical knowledge for brain tumor diagnosis.

3 Algorithms

3.1 Iterative I-Projections

MECore’s default inference algorithm is based on the idea of Iterative I-Projections [4]. Given a probability distribution \( P \) and a consistent knowledge base \( R \), the conditionals \( c \in R \) are traversed cyclically in an arbitrary but fixed order. In

\(^1\) Both test sets can be downloaded at http://www.fernuni-hagen.de/wbs/data/mecore_testsets_dkb_2013.zip
each such step the current probability distribution $P_i$ is I-projected onto $\text{Mod}(c)$ yielding the next probability distribution $P_{i+1}$. Starting with $P_1 = P$ this process converges to the I-projection of $P$ onto $\text{Mod}(\mathcal{R})$ [4]. If no prior distribution is given, the algorithm starts with the uniform distribution and converges to the distribution that satisfies $\mathcal{R}$ and maximizes entropy.

To understand MECore’s inference algorithm in more detail, consider a knowledge base $\mathcal{R} = \{ (\psi_1 \mid \phi_1)[x_1], \ldots, (\psi_m \mid \phi_m)[x_m] \}$ and let $P^*$ denote the I-projection of $P$ onto $\text{Mod}(\mathcal{R})$. One can show that there is a real number $a_i$ for each conditional $(\psi_i \mid \phi_i)[x_i] \in \mathcal{R}$ and a normalizing constant $a_0$ such that for each world $\omega \in \Omega$ with positive probability $P^*(\omega) > 0$ the distribution $P^*$ factorizes in the following way [4]:

$$P^*(\omega) = a_0 P(\omega) \prod_{i=1}^{n} a_i^{(1_{\{ \psi_i \phi_i \}}(\omega) \cdot (1-x_i) - 1_{\{ \neg \phi_i \}}(\omega) \cdot x_i)}.$$  

(2)

Note that the exponent of $a_i$ corresponds to the factor of $P(\omega)$ of the constraint function of the $i$-th conditional in (1). An intuitive interpretation of the factorization is given in [13]: We say a world $\omega \in \Omega$ verifies the conditional $(\psi_i \mid \phi_i)[x_i]$ iff $1_{\{ \psi_i \phi_i \}}(\omega) = 1$, i.e., iff it satisfies the antecedence and the consequence of the conditional. We say $\omega$ falsifies the conditional iff $1_{\{ \neg \phi_i \}}(\omega) = 1$, i.e., iff it satisfies the antecedence but falsifies the consequence of the conditional. Hence, each conditional has a positive effect $a_i^{(1-x_i)}$ on the probability of the world, if the world verifies the conditional and a negative effect $a_i^{-x_i}$ if the world falsifies the conditional. One can also show that whenever there is a distribution that factorizes according to (2), it is necessarily the I-Projection of $P$ onto $\text{Mod}(\mathcal{R})$ [4].

Note that if all probabilities are positive, the complete distribution $P^*$ factorizes according to (2) and belongs to the following exponential family

$$\{ Q : \Omega \to [0,1] \mid Q(\omega) = c_0 P(\omega) \prod_{i=1}^{n} c_i^{(1_{\{ \psi_i \phi_i \}}(\omega) \cdot (1-x_i) - 1_{\{ \neg \phi_i \}}(\omega) \cdot x_i)} , Q(\Omega) = 1 \}.$$  

If $P^*$ is non-positive, it still belongs to the topological closure of this family [5]. Hence, even if the I-projection $P^*$ of $P$ onto $\text{Mod}(\mathcal{R})$ does not factorize, there is a sequence in the exponential family that converges to $P^*$. In this case, some factors in (2) might tend to infinity, but usually relatively small factors are sufficient for a good approximation, as we illustrate later on. Instead of computing the I-projection of $P$ onto $\text{Mod}(\mathcal{R})$ by iterative I-projections, MECore computes the I-projection onto the exponential family.

**Computing I-projections by scaling:** In [20] a simple update formula is developed to compute the sequence of I-projections onto the exponential family. Basically, whenever I-projecting the current distribution $P_i$ onto $\text{Mod}(c)$ for a conditional $c \in \mathcal{R}$ a scaling factor $\beta_i$ is computed to adapt the factor $a_c$ in the factorization of $P_i$ appropriately. Then the factorization of the I-projection $P_{i+1}$
is obtained from $P_i$ by multiplying $a_c$ by $\beta_i$. The computational cost for one iteration is linear in the number of worlds $|\Omega|$. It is unnecessary to renormalize the factorization in each step, since probabilities of intermediate distributions are considered in a way that the normalization constant cancels out. Hence $a_0$ is computed after termination.

MECore’s inference algorithm is based on this update formula. If no knowledge is available, i.e., if $\mathcal{R} = \emptyset$, the initial distribution is the uniform distribution and its factorization is $P(\omega) = a_0$, where $a_0 = \frac{1}{|\Omega|}$. If the distribution has to be adapted to new conditionals $c_1, \ldots, c_n$, a new factor $a_i$ for each conditional is introduced and is initialized with 1. Then iteratively the I-projections of the current distribution onto $\text{Mod}(c_i)$ are computed by applying the update method of [20], i.e., by successively scaling the factors $a_{c_i}$ with a factor $\beta_i$. As explained before, the process converges to the I-projection onto $\text{Mod}(\mathcal{R})$. See [20] for a more thorough discussion.

An easy computable abortion criterion for the iterative scaling process is the relative change of the adapting factors $\beta_i$. If they are close to 1 for all conditionals the algorithm stops. If, however, it is noticed that the factors tend to infinity, the relative change in the actual probabilities has to be computed to avoid divergence.

**Example 1.** Consider the (pathological) knowledge base $\{(AB)[0.6], (A\overline{B})[0.4]\}$ over the signature $\Sigma = \{A, B\}$ that enforces probability 0.6 and 0.4 respectively for two worlds and therefore enforces probability 0 for the remaining two worlds. As the factors for the two conditionals grow, the distribution is approximated better and better, yet is never obtained. However, as Table 2 indicates, comparatively small factors are sufficient for a good approximation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>$P(AB)$</th>
<th>$P(A\overline{B})$</th>
<th>$P(\overline{A}B)$</th>
<th>$P(\overline{A}B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.31</td>
<td>0.32</td>
<td>0.32</td>
<td>0.62511</td>
<td>0.36268</td>
<td>0.00847</td>
<td>0.00847</td>
</tr>
<tr>
<td>0.30</td>
<td>0.936</td>
<td>0.896</td>
<td>0.59880</td>
<td>0.40046</td>
<td>0.00051</td>
<td>0.00051</td>
</tr>
<tr>
<td>0.31</td>
<td>19.20</td>
<td>18.80</td>
<td>0.60000</td>
<td>0.40001</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Table 2. Factors and induced probabilities for $c_1 = (AB)[0.6], c_2 = (A\overline{B})[0.4]$.

### 3.2 L-BFGS

Formally, computing I-projections is a convex optimization problem, see, e.g., [10] or [23] for a comprehensive introduction to numerical optimization. Non-commercial standard solvers are usually outperformed by iterative scaling variants, as they rely on general techniques to maintain the constraints and do not take the special structure of the problem into account. To obtain an unconstrained optimization problem that can be solved more efficiently, we can consider the Wolfe dual of relative entropy minimization [10]. We restrict ourselves to practical aspects here, and refer to [26] for technical details.
As the Wolfe dual yields a convex objective function on a convex set, we can apply line search methods to compute the minimum. Gradient descent is one particular prominent example, which, however, can converge quite slowly as it often moves along a 'zigzag' path instead of a smooth path towards the solution [23]. There are more sophisticated gradient-based methods like conjugate gradient, which provide better convergence properties, but in the context of exponential models they are often outperformed by algorithms taking second-order derivatives into account [19,21]. In general, second-order derivatives are hard to compute and expensive to store, therefore limited memory variable metric methods approximate these derivatives and represent the information implicitly [23].

Limited Memory BFGS (L-BFGS) [17] is a member of the family of limited memory variable metric methods. When computing parameter estimations for different exponential models, iterative scaling algorithms are often outperformed by such algorithms, [19,21]. L-BFGS has become popular among these algorithms and is also used in the context of probabilistic reasoning, see, e.g., [28,11]. In [26] L-BFGS is applied to compute I-projections for relational conditional languages and a significant speedup compared to two recent iterative scaling implementations from [7] is obtained. At the moment, the L-BFGS implementation of the RISO-project \(^2\) is being integrated into MECore to evaluate its performance on propositional conditional languages.

As before, we want to compute the I-projection \(P^*\) of a prior distribution \(P\) onto the set of probability distributions that satisfy a knowledge base \(R = \{(\psi_1|\phi_1)[x_1],\ldots,(\psi_m|\phi_m)[x_m]\}\). When deriving factorization (2) by using Lagrangian methods, the factors \(a_i\) corresponding to the \(i\)-th conditional have the form

\[
a_i = \exp(\lambda_i),
\]

where \(\lambda_i\) is a Lagrange multiplier. If unfamiliar with Lagrangian methods, \(\lambda_i\) can be simply regarded as a weight determining the influence of the \(i\)-th conditional on the probability distribution. The normalizing factor becomes \(a_0 = \exp(\lambda_0 - 1)\).

Replacing the \(a_i\) in equation (2) with their exponential counterparts yields a function \(h_\omega : \mathbb{R}^{n+1} \to \mathbb{R}\) for each \(\omega \in \Omega\) that maps a vector of \(n + 1\) multipliers \(\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_n)\), to a real number

\[
h_\omega(\lambda) = P(\omega) \cdot \exp(\lambda_0 - 1 + \sum_{i=1}^{n} \lambda_i \cdot (1_{\{\psi_i|\phi_i\}(\omega)} \cdot (1 - x_i) - 1_{\{\psi_i|\phi_i\}(\omega)} \cdot x_i)).
\]

Note that for the vector of Lagrange multipliers \(\lambda^*\) it holds \(h_\omega(\lambda^*) = P^*(\omega)\) if \(P^*(\omega) > 0\). As before, we consider a sequence of exponential families that converges to the I-projection of \(P\) onto \(R\). But whereas iterative scaling algorithms scale each factor in turn, L-BFGS applied to the Wolfe dual adapts the corresponding multipliers \(\lambda\) simultaneously by taking first-order and approximated second-order derivatives into account.

\(^2\) http://sourceforge.net/projects/riso/
The dual problem results as follows [26]:

$$\min_{\lambda \in \mathbb{R}^{n+1}} -\lambda_0 + \sum_{\omega \in \Omega} h_\omega(\lambda)$$  \hspace{1cm} (5)

Each component of the gradient of the objective function (5) corresponds to the error of a constraint function. The first component corresponding to the normalizing multiplier $\lambda_0$ is

$$\sum_{\omega \in \Omega} h_\omega(\lambda) - 1.$$  \hspace{1cm} (6)

That is, it is 0 iff the values $h_\omega$ sum to 1. As $\exp(x) > 0$ for all $x \in \mathbb{R}$ it assures that the solution yields a probability distribution. The component corresponding to the multiplier $\lambda_i$ of the $i$-th conditional, $1 \leq i \leq n$, is

$$\sum_{\omega \in \Omega} h_\omega(\lambda) \cdot (1_{\{\psi\phi\}}(\omega) \cdot (1 - x) - 1_{\{\overline{\psi}\phi\}}(\omega) \cdot x)$$  \hspace{1cm} (7)

Note that it corresponds to the constraint function of the $i$-th conditional shown in equation (1), where $P(\omega)$ is replaced with $h_\omega$. Therefore the component is 0 iff the $i$-th conditional is satisfied. Hence the optimum is indeed a probability distribution that approximately satisfies $\mathcal{R}$. L-BFGS terminates as soon as the Euclidean norm of the gradient is smaller than a predefined $\epsilon > 0$, i.e., if the root of the squared error of all constraints is lower than $\epsilon$. Recalling the remark that a distribution that factorizes according to (2) is necessarily the I-Projection of $\mathcal{P}$ onto $\text{Mod}(\mathcal{R})$ [4], it is intuitively clear that the algorithm indeed converges to the I-projection of $\mathcal{P}$ onto $\text{Mod}(\mathcal{R})$. A more rigorous discussion of the technical details can be found in [26].

### 3.3 Comparison and Discussion

We compared runtime results for MECores Iterative I-projection (IIP) implementation and L-BFGS for the knowledge bases shown in Table 1. Our results are shown in Table 3.

<table>
<thead>
<tr>
<th>Name</th>
<th>Time IIP</th>
<th>Error IIP</th>
<th>Time L-BFGS</th>
<th>Error L-BFGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV</td>
<td>4.9</td>
<td>$\approx 10^{-3}$</td>
<td>38.7</td>
<td>$\approx 10^{-3}$</td>
</tr>
<tr>
<td>CarDiagnosis</td>
<td>51.0</td>
<td>$\approx 10^{-2}$</td>
<td>143.0</td>
<td>$\approx 10^{-2}$</td>
</tr>
<tr>
<td>BrainTumor</td>
<td>9.8</td>
<td>$\approx 10^{-3}$</td>
<td>15.2</td>
<td>$\approx 10^{-3}$</td>
</tr>
</tbody>
</table>

**Table 3.** Runtime comparison in seconds: Iterative I-projections vs. L-BFGS.

To assure comparability, we configured both algorithms to keep an error bound measured by the Euclidean distance. IIP provided in all experiments a
slightly lower error while simultaneously being faster. This is remarkable since usually iterative scaling algorithms are clearly outperformed by L-BFGS [19,21], also in the context of computing I-projections with respect to conditional knowledge bases [7]. However, these iterative scaling implementations are based on another update formula from [6] and its improved formula from [3]. For distinction, we call these formulas GIS-updates and we call MECore’s tailor-made update formula from [20] the IIP-update. As introducing these update formulas requires some additional background information and notation, we cannot discuss the differences here and have to refer the interested reader to the cited literature.

It seems that the IIP-update provides significantly better convergence properties than GIS-updates. Note that even though the improved GIS-update from [3] also decreases the number of iterations, often the overall runtime of the algorithm is not improved but even increased as the computational cost for each iteration is increased significantly [19]. In contrast, MECore’s update formula preserves the computational cost for a single iteration. Therefore an interesting question for future work is, to which extent the formula can be generalized to non-propositional knowledge bases. However, it is possible that in this case no comparatively simple closed formula exists.

4 Optimization of Algorithms

A crucial influencing factor for the runtime of each of the mentioned algorithms is the number of worlds $|Ω|$. In each iteration all worlds have to be traversed to compute a scaling update and also to compute the objective function and the gradient for L-BFGS. To decrease the number of worlds $|Ω|$ currently two preprocessing steps are being integrated into MECore, which we describe in the following.

4.1 Exploiting Deterministic Knowledge

Conditionals $(ψ|φ)[x]$ having probability 0 or 1 basically enforce zero-probabilities. If $x = 0$ the corresponding constraint (1) becomes $\sum_{ω ∈ Ω} p(ω) \cdot 1_{\{ψφ\}}(ω) = 0$. As each probability is non-negative, each world verifying the conditional, i.e., $1_{\{ψφ\}}(ω) = 1$, has probability zero if $P$ satisfies the conditional. Symmetrically, for $x = 1$ the corresponding constraint in (1) becomes $\sum_{ω ∈ Ω} p(ω) \cdot 1_{\{¬ψφ\}}(ω)$. Hence each world falsifying the conditional has probability zero if $P$ satisfies the conditional. So for each $P$ satisfying a knowledge base, each world verifying a 0-conditional and each world falsifying a 1-conditional has necessarily probability 0 and therefore does not has to be regarded when solving the optimization problem.

Therefore, in a preprocessing step, the deterministic conditionals $R^{det}$ are removed from $R$ and the corresponding zero-worlds $N ⊆ Ω$ are deleted in $R$. The computational cost is $O(|R^{det}| : |Ω|)$, if simply traversing the worlds and removing those ones that verify a 0-conditional or falsify a 1-conditional. For sufficiently complex problems, the preprocessing costs are negligible regarding
the overall cost of the algorithms. If many zero probabilities are determined by deterministic conditionals one can obtain a significant speedup.

To complete the factorization in (2) for the deterministic conditionals, the factors corresponding to deterministic conditionals can be defined as $0^1(\psi, \phi, \omega)$ if $x = 0$ and $0^1(\omega^x, \omega)$ if $x = 1$, when defining $0^0 = 1$. This definition is in accordance with both algorithms, since the Lagrange multipliers of deterministic conditionals tend necessarily to $\pm \infty$ and the corresponding factors $a$ in the factorization (2) tend to zero. See [27] for a more thorough discussion.

### 4.2 Exploiting Equivalence of Worlds

As explained before, worlds can verify or falsify conditionals. This observation gives rise to the conditional structure of worlds [13], which can be exploited to speed up intermediate computation steps significantly [7]. Given a knowledge base $\mathcal{R}$, the conditional structure $\sigma_{\mathcal{R}}(\omega)$ of $\omega \in \Omega$ can be defined as an $|\mathcal{R}|$-tuple, over abstract symbols $\{+, -, 0\}$, i.e., $\sigma_{\mathcal{R}} : \Omega \rightarrow \{+, -, 0\}^{|\mathcal{R}|}$. The $i$-th component of the tuple is $+$ if $\omega$ verifies the $i$-th conditional, $-$ if $\omega$ falsifies the conditional, and 0 otherwise.

**Example 2.** Consider the knowledge base $\mathcal{R} = \{(B|A)[0.8], (C|A)[0.2]\}$ over the signature $\Sigma = \{A, B, C\}$. As it is convenient, we represent worlds by complete conjunctions $A^a B^b C^c$, where $a, b, c \in \{0, 1\}$ and $a^0 := \overline{a}$, $a^1 := a$ for each $a \in \Sigma$. The complete conjunction $A^a B^b C^c$ corresponds to the interpretation that assigns the truth value $a$ to $A$, $b$ to $B$ and $c$ to $C$. The conditional structure for each world with respect to our knowledge base $\mathcal{R}$ is a 2-tuple from $\{+, -, 0\}^2$. For instance

$$\sigma_{\mathcal{R}}(ABC) = (+, +),$$
$$\sigma_{\mathcal{R}}(\overline{A}BC) = (0, 0),$$
$$\sigma_{\mathcal{R}}(\overline{A}\overline{B}C) = (-, +).$$

If two worlds have the same conditional structure, they necessarily have the same probability in the I-projection onto $\text{Mod}(\mathcal{R})$, since their probabilities are generated by the same factors in factorization (2). As pointed out in [7], many worlds have the same conditional structure and it can be very beneficial to exploit this redundancy.

Consider the equivalence relation $\equiv_{\mathcal{R}} \subseteq \Omega \times \Omega$ corresponding to the conditional structure of worlds, i.e., $\omega_1 \equiv_{\mathcal{R}} \omega_2$ iff $\sigma_{\mathcal{R}}(\omega_1) = \sigma_{\mathcal{R}}(\omega_2)$. The equivalence classes $\Omega / \equiv_{\mathcal{R}}$ can be computed with computational cost $O(|\mathcal{R}| \cdot |\Omega|)$ by traversing the worlds and partitioning them with respect to their conditional structure. Whenever a function

$$\sum_{\omega \in \Omega} P(\omega) \cdot f(\omega)$$

is evaluated,
has to be computed for some function $f : \Omega \rightarrow \mathbb{R}$, one can replace this equation by

$$\sum_{[\omega] \in \Omega/\equiv} |[\omega]| \cdot P(\omega) \cdot f(\omega), \quad (9)$$

where $|[\omega]|$ denotes the size of the equivalence class $[\omega]$.

For example, if $P$ is unnormalized, the reciprocal of the normalizing factor $\frac{1}{a_0} = \sum_{\omega \in \Omega} P(\omega)$ can be computed by (9), where $f$ is the constant function 1, i.e., $\frac{1}{a_0} = \sum_{[\omega] \in \Omega/\equiv} |[\omega]| \cdot P(\omega)$. Note that the representative $\omega$ of $[\omega]$ can simply be represented by the corresponding $\{+, -, 0\}$-tuple $\sigma(\omega)$, since it contains all information that is needed to compute the probability $P(\omega)$ with factorization (2). As for preprocessing of deterministic conditionals, the preprocessing costs are negligible regarding the overall cost of the algorithms for sufficiently complex problems.

### 4.3 Comparison and Discussion

Both optimizations have been implemented for MECore’s Iterative I-projection (IIP) implementation. We compared runtime results for the original and the optimized version of IIP for the knowledge bases shown in Table 1. The results are shown in Table 4. As the runtime for the optimized version can be broken down into preprocessing time and the time that is actually used for computing the I-projections, the last column shows the time spend for both steps separately.

<table>
<thead>
<tr>
<th>Name</th>
<th>Time IIP</th>
<th>Time Optimized IIP</th>
<th>Optimized preprocessing/ IIP time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV</td>
<td>7.9</td>
<td>44.9</td>
<td>33.7/11.2</td>
</tr>
<tr>
<td>CarDiagnosis</td>
<td>284.0</td>
<td>134.8</td>
<td>88.3/46.5</td>
</tr>
<tr>
<td>BrainTumor</td>
<td>54.7</td>
<td>28.7</td>
<td>9.8/18.9</td>
</tr>
</tbody>
</table>

**Table 4.** Runtime comparison in seconds: Original vs. Optimized IIP.

If comparing the runtime results to Table 3 from the last section, one might notice that the original IIP-algorithm performs slower. The reason is that the accuracy of both algorithms has been increased. We do not compare the error here, as both algorithms use the same termination criterion. There are slight differences in the error, which, however, result from numerical reasons, as the probabilities are computed in a slightly different way.

Table 4 shows that the original version of the algorithm is faster for the TV knowledge base, but is outperformed by its optimized counterpart for the other two knowledge bases. The reason is that to compensate additional costs for preprocessing and a different internal world representation, the knowledge base has to be sufficiently complex.
5 Conclusion and Future Work

We gave an overview of algorithmic approaches applied in the expert system shell MECore [8]. As we saw in Section 3, MECores IIP-update formula [20] can compete with L-BFGS, even though the latter usually clearly outperforms iterative scaling algorithms based on the GIS-update from [6]. Therefore, we are currently investigating whether a similar simple closed formula can be defined for the class of linearly structured semantics [26], which captures relational languages and their semantics [15,9] in addition to the propositional framework considered in this work.

In Section 4, we sketched two methods that can be applied to exploit the structure of conditional knowledge bases. The first method exploits the determinism of deterministic conditionals, the second method exploits redundancies regarding the conditional structure of worlds [7]. Whereas the benefit of the former method is restricted to knowledge bases containing deterministic conditionals, the second method provides a general speedup provided that the problem domain is sufficiently complex. However, for small problems both methods can increase the runtime due to preprocessing time and more complex data structures. In future work, we plan to integrate both methods in our L-BFGS implementation.

Note that the problem we were dealing with here is to compute a probability distribution to establish an epistemic state. To the best of our knowledge, this problem has no immediate counterpart in other probabilistic frameworks. It is related to learning Markov Random Fields [16,12], where, however, the factors are adapted with respect to statistical datasets, whereas they are adapted with respect to conditional knowledge bases in our framework. In [24] a more closely related method is introduced to learn Markov Logic Networks [28] from both statistical data and expert knowledge. We did not provide a runtime comparison between both methods as both problems can be solved by the same algorithms. As the corresponding exponential families differ in their structure, algorithms can be optimized in different ways, but a discussion of this topic is out of the scope of this paper.

An important problem that we did not regard in this work is the computation of probabilities of (conditional) formulas. Up to now in MECore a table-based approach is applied, where all worlds are traversed and their probabilities are summed up depending on whether they satisfy or falsify a formula. As we consider only I-projections onto exponential families, which basically correspond to Markov Random Fields, it is an interesting task for future work to integrate inference techniques for graphical models [16] into MECore.

Another important issue is the consistency of knowledge bases. In particular when combining knowledge from different sources inconsistencies arise easily. In [30] inconsistency measures are considered and a method is proposed to map arbitrary conditional knowledge bases to consistent ones by minimally changing the conditionals’ probabilities. Integrating such methods into MECore is desirable since resolving inconsistencies manually is a difficult and time-consuming task.
References


Abstract. Systems of logico-probabilistic (LP) reasoning characterize inference from conditional assertions that are interpreted as expressing high conditional probabilities. In previous work, we studied four well known LP systems (namely, systems O, P, Z, and QC), and presented data from computer simulations in an attempt to illustrate the performance of the four systems. These simulations evaluated the four systems in terms of their tendency to license inference to accurate and informative lower probability bounds, given incomplete information about a randomly selected probability distribution (where this probability distribution may understood as representing the true stochastic state of the world). In our earlier work, the procedure used in generating the unknown probability distribution (i.e., the true stochastic state of the world) tended to yield probability distributions with moderately high entropy levels. In the present article, we present data charting the performance of the four systems in reasoning about probability distributions with various entropy levels. The results allow for a more inclusive assessment of the reliability and robustness of the four LP systems.

Keywords: ampliative inference, default reasoning, non-monotonic reasoning, probability logic.

1 LP Reasoning: Systems O, P, Z, and QC

We represent the four LP systems considered here (O, P, Z, and QC; described below) using a simple propositional language L, with the usual connectives \( \neg, \land, \lor, \) and \( \supset \), and A, B, C, etc. as meta-logical variables ranging over arbitrary sentences of L. Our main interest will be in extensions of L by means of a default (or uncertain) conditional operator: \( \Rightarrow \). In particular, we will concern ourselves with extensions of L by simple uncertain conditionals of the form A\( \Rightarrow \)B. Throughout the paper, \( \alpha \) and \( \beta \) will serve as meta-variables ranging over such simple conditional formulas, while \( \Gamma \) ranges over sets of them. “\( \vdash \)” is used to denote derivability in classical logic, and “\( \bot \)” to denote an arbitrary contradiction. The four LP systems that we consider are ordered in terms of the number of inferences they license (\( O \subset P \subset Z \subset QC \)). We proceed by considering the weakest system first.
1.1 System O

System O is of interest because of its close connection to the following consequence relation:

\[(1) \text{ Strict Preservation: } A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n \quad \text{s.p.} \quad C \Rightarrow D \quad \text{iff for all probability functions } P \text{ (over } L): P(D|C) \geq \min\{P(B_i|A_i) : 1 \leq i \leq n\}.\]

System O was developed by Hawthorne [1] and Hawthorne and Makinson [2] as an inferential calculus for \(\text{|| s.p.}\). Throughout the present article, ”\(-\text{O}\)” denotes the syntactical notion of derivability in system O.

**System O** (after Hawthorne):

- **REF** (reflexivity): \(\Gamma \text{||O} A \Rightarrow A\)
- **LLE** (left logical equivalence): if \(\Gamma \text{||O} (A \Rightarrow B) \land (B \Rightarrow A)\), then \(\Gamma \text{||O} A \Rightarrow C\)
- **RW** (right weakening): if \(\Gamma \text{||O} B \Rightarrow C\), then \(\Gamma \text{||O} A \Rightarrow B\)
- **VCM** (very cautious monotony): \(\Gamma \text{||O} A \Rightarrow B \land C\)
- **XOR** (exclusive Or): if \(\Gamma \text{||O} \lnot (A \land B)\), then \(\Gamma \text{||O} A \Rightarrow C, B \Rightarrow C\)
- **WAND** (weak And): \(A \Rightarrow B, A \land \lnot C \Rightarrow \bot \text{||O} A \Rightarrow B \land C\)

It is easy to see that all of the rules of system O are correct with respect to \(\text{|| s.p.}\), i.e., \(\Gamma \text{||O} A \Rightarrow B\) implies \(\Gamma \text{|| s.p.} A \Rightarrow B\). It was the hope of Hawthorne and Makinson [2] that \(\text{||O}\) was also complete with respect to \(\text{|| s.p.}\), but Paris and Simmonds [3] have shown that this not the case.

Following [5], we propose a marriage of system O, and a rule for inferring lower probability bounds that corresponds to the correctness of system O for \(\text{|| s.p.}\). To make sense of such inferences, we employ statements of the form “\(A \Rightarrow, B\)” to express that \(P(B|A) \geq r\), and say that system O licenses the (valid) inference to \(\text{||O} C \Rightarrow \text{|| s.p.} D\) from \(A_1 \Rightarrow, B_1, \ldots, A_n \Rightarrow, B_n\) in cases where \(A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n \text{||O} C \Rightarrow D\).

A remarkable fact about O is its weakness compared to standard systems of conditional logic. According to Segerberg [4], the weakest ‘reasonable’ system of conditional logic includes REF, LLE, and RW, along with the following rule:

\[(\text{AND}): \text{from } A \Rightarrow B \text{ and } A \Rightarrow C \text{ infer } A \Rightarrow B \land C.\]

The inferential power of AND is quite significant. By adding AND to the system O, we obtain (in one step) the well known system P. In comparison with WAND, we note that \(A \Rightarrow C\) is derivable from \(A \land \lnot C \Rightarrow \bot\), given RW, REF, and XOR, while
A∧¬C⇒⊥ is not derivable from A⇒C, given these rules. It is in this respect that WAND is weaker than the rule AND.  

1.2 System P

As described in [5], system P represents the confluence of a number of different semantic criteria. But the feature of system P that is of greatest interest here is its connection with the following consequence relation (cf. [6]):

(2) Improbability-Sum Preservation: A_1⇒B_1, ..., A_n⇒B_n ├_{i.s.p.} C⇒D iff for all probability functions over L: I(D|C) ≤ Σ{I(B_i|A_i) : 1≤i≤n}, where I(A|B) is defined as 1−P(A|B).

Adams demonstrated that the following calculus (denoted by ├_{i.s.p.}') is correct and complete for ├_{i.s.p.}.

System P (after Adams):

REF
LLE
{as with system O
RW
AND: as above
CC (cautious cut): A⇒B, A∧B⇒C ├_{i.s.p.} A⇒C
CM (cautious monotony): A⇒B, A⇒C ├_{i.s.p.} A∧B⇒C
OR: A⇒C, B⇒C ├_{i.s.p.} A∧B⇒C

Following [5], we propose a marriage of system P, and a rule for inferring lower probability bounds that corresponds to the correctness of system P for ├_{i.s.p.}.' In particular, we say that system P licenses the (valid) inference to C⇒1−Σ_{1≤i≤n} D from A_1⇒_{i.s.p.} B_1, ..., A_n⇒_{i.s.p.} B_n, in cases where A_1⇒B_1, ..., A_n⇒B_n ├_{i.s.p.} C⇒D.

1.3 System Z

While system P sanctions more inferences than system O, it still sanctions fewer inferences than one might reasonably accept. For instance, P does not licence inference via subclass inheritance based on default assumptions of irrelevance (or independence). For example, if we know that this animal is a male bird (B∧M) and that birds can normally fly (B⇒F), and nothing else of relevance, then we would intuitively draw the conclusion that this male bird can fly (F). However, B∧M⇒F is not P-

---

1 We also observe that XOR is weaker than the rule OR (introduced below), and that VCM implies CM (below), in the presence of AND, while CM implies VCM (below), in the presence of RW ([2], 251).
entailed by B⇒F, because there are possible probability distributions in which P(F|B∧M) is much smaller than P(F|B). If we do infer B∧M⇒F from B⇒F, in such cases, then we assume, by default, that the additional factor M (in this case the gender of a bird) is irrelevant to its ability to fly (or in other words, M and F are assumed to be probabilistically independent given B). A straightforward means of enlarging the set of LP-derivable conditionals, in order to include such default inferences, is to give up the requirement that a reasonable inference be valid for all possible probability distributions, and consider only ‘normal’ probability distributions, i.e., those distributions which satisfy the default assumption of irrelevance. An early suggestion for realizing this idea was the maximum entropy approach to default inference (cf. [7]; [8], 491-3). By selecting a probability distribution that maximizes entropy, one minimizes probabilistic dependences. Despite having some attractive features, the maximum entropy approach is rather complicated, and has some further disadvantages, such as language dependence.

System Z of Pearl [9] and Goldszmidt and Pearl [10] maintains many of the advantages of the maximum entropy approach, while overcoming its disadvantages. Like the maximum entropy approach, inference in system Z proceeds via the construction of a semantic model of the premise conditionals that maximizes probabilistic independences. In system Z, this is achieved by maximizing the degree-of-normality of the set of possible worlds represented by a ranked model, according to the following definition:

(3) Definition (cf. [10], 68, def. 15; [11], 308f): A ranked model (W, r) is as least as normal as a ranked model (W, r*) (with the same world set), in short (W, r) ≥N (W, r*), iff for all w∈W, r(w) ≤ r*(w).

As has been shown (cf. [5]; [9]), every set of worlds W (which is constructed over the language of the conditional knowledge base Γ) has a unique most normal ranked model, the so called z-model. In order to define the notion of a z-model, we first define the notion of a z-rank.

(4) Definition ([9], section 1; [10], 65, fig. 2): For every (finite) P-consistent\(^2\) set of conditionals Γ = {A₁⇒B₁,..., Aₙ⇒Bₙ}, the z-rank of the elements of Γ is defined by the following z-algorithm:

(i) Initial step: Set i = 0. Set Δ = Γ.

(ii) Iterative step:

1. If Δ is nonempty, let Δᵢ ⊆ Δ consist of all conditionals α in Δ which are tolerated by Δ, otherwise go to (iii).\(^3\)

2. If Δᵢ is nonempty, set Δ = Δ−Δᵢ, and i = i +1.

3. If Δᵢ is empty, set Δ∞ = Δ, and set Δ = ∅.

\(^2\) A set of conditionals Γ is called P-consistent iff Γ does not P-entail ¬⊥⇒⊥.

\(^3\) A conditional A⇒B is tolerated by Δ, if there is a possible world over the propositional atoms appearing in Δ that verifies A∧B and does not falsify any conditional in Δ.
The z-partition \((\Delta_0, \ldots, \Delta_k, \Delta_\infty)\).

The z-rank of a conditional \(\alpha\) in a \(P\)-consistent \(\Gamma\), written “\(z_{\Gamma}(\alpha)\)”, is defined as the index \(i\) of that set \(\Delta_i\) in the z-partition of \(\Gamma\) in which \(\alpha\) occurs.

The assumption of the preceding definition, that \(\Gamma\) is \(P\)-consistent, guarantees that there is a z-model for \(\Gamma\), according to the following definition:

(4) Definition ([9], 123-5, Eq. 5, 6, and 10): The z-model of a \(P\)-consistent \(\Gamma\), \((W_{\Gamma}, z_{\Gamma})\), is defined as follows:

For each \(w\) among the set of logically possible worlds over the propositional atoms appearing in \(\Gamma\):

(i) If \(w\) falsifies \(\Delta_\infty\), then \(w \notin W_{\Gamma}\).

Else: (ii) \(w \in W_{\Gamma}\), and \(z_{\Gamma}(w) = 0\), if \(w\) doesn’t falsify any \(\alpha\) in \(\Gamma\); otherwise \(z_{\Gamma}(w) = \max(\{ z_{\Gamma}(\alpha) : w\) falsifies \(\alpha\}) + 1\).

(iii) The z-rank of an arbitrary formula \(C\) relative to \((W_{\Gamma}, z_{\Gamma})\) is defined as \(z_{\Gamma}(C) = \min(\{ z_{\Gamma}(w) : w\) \in \(W_{\Gamma}\) and \(w\) verifies \(C\})\), with \(\min(\emptyset) = \infty\).

(iv) For all \(\Gamma\): \(\Gamma \models \sim\sim Z C \Rightarrow D (\Gamma Z\text{-entails } C \Rightarrow D)\) iff either (a) \(\Gamma\) is \(P\)-inconsistent, or (b) \(C \Rightarrow D\) is satisfied in \((W_{\Gamma}, z_{\Gamma})\) (i.e., all worlds with rank \(z_{\Gamma}(C)\) verify \(D\)).

Z-entailment validates inference by default inheritance (i.e., \(A \Rightarrow B \models \sim\sim Z A \land C \Rightarrow B\)) as well as default contraposition (i.e., \(A \Rightarrow B \models \sim\sim Z \neg B \Rightarrow \neg A\)). That these inferences hold ‘by default’ means that they hold under the condition that the conditional knowledge base doesn’t contain further conditionals that are \(\varepsilon\)-inconsistent with the conclusions of these inferences (cf. Adams 1975). The relation \(\models \sim\sim Z\) is thus non-monotonic, since, for example, whether \(\Gamma \cup \{A \Rightarrow B\} \models \sim\sim Z \neg B \Rightarrow \neg A\), depends on whether \(\Gamma \cup \{A \Rightarrow B\} \cup \{\neg B \Rightarrow \neg A\}\) is \(\varepsilon\)-inconsistent.

One disadvantage of Z-entailment is that (in the absence of further assumptions) it does not automatically provide information concerning probabilistic reliability, such as provided by the improbability-sum semantics for system \(P\). However, in [5] it is shown how to obtain this desideratum (based on work in [12]):

**Theorem 1** If \(A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n \models \sim\sim Z C \Rightarrow D\) holds, then improbability-sum preservation \((I(D|C) \leq \Sigma I(B_i|A_i) : 1 \leq i \leq n))\) holds for all probability functions \(P\) that satisfy the default assumptions \(P(A_i \Rightarrow B_i|C) \geq P(B_i|A_i), \text{ for all } 1 \leq i \leq n\).

Proof: See [5], theorem 4 (5).

We proceed here as if the default assumptions specified in theorem 1 hold, and say that system Z licenses the inference to \(C \Rightarrow_{1-\Sigma} D\) from \(A_1 \Rightarrow_{1-\gamma} B_1, \ldots, A_n \Rightarrow_{1-\gamma} B_n\) in cases where \(A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n \models \sim\sim Z C \Rightarrow D\). As with the evaluations conducted in

\(^4\) A set of conditionals is \(\varepsilon\)-consistent just in case the corresponding conditional probabilities can be simultaneously made arbitrarily close to 1.
[5], a central question concerns whether inference in accordance with the preceding principle tends to yield accurate conclusions.

1.4 System QC

η-entailment is not the strongest (minimally reasonable) inference calculus for ‘risky’ default inference among uncertain conditionals. An even stronger and extremely simple calculus is quasi-classical reasoning. Here one reasons with uncertain conditionals as if they were material implications:

\[(5) \Gamma \vdash_{\text{QC}} C \Rightarrow D \iff \{ A \Rightarrow B : A \Rightarrow B \in \Gamma \} \vdash C \Rightarrow D.\]

Improbability-sum preservation holds for inferences between material conditionals, or more generally, between formulas of propositional logic, as was shown by Suppes ([13], 54). In particular, \(\{A_1, \ldots, A_n\} \vdash B\) iff it holds for all probability distributions that \(I(B) \leq \Sigma \{I(A_i):1 \leq i \leq n\}\). Beyond the result of Suppes, it is possible to formulate probabilistic conditions under which QC-reasoning approximately satisfies improbability-sum preservation. In particular, it is shown in ([5], sec. 2.5, (13)) that a QC inference from a given set of premises is guaranteed to preserve probability in the manner of system P iff the improbability-sum of the premises is very small, and some decimal powers smaller than the probability of the conclusion’s antecedent. Following [5], we proceed as if these conditions hold, and say that system QC licenses the inference to \(C \Rightarrow 1 - \Sigma \{1 - r_i : 1 \leq i \leq n\}\) from \(A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n\), in cases where \(A_1 \Rightarrow B_1, \ldots, A_n \Rightarrow B_n \vdash_{\text{QC}} C \Rightarrow D\). The question remains of whether inference in accordance with the preceding principle tends to yield accurate conclusions.

2 The Simulations

Following [5], our simulations operate over a simple language with four two-valued variables: a, b, c, and d. Similarly, we assume a probability distribution over the sixteen possible worlds describable in this language. For all of our simulations, we generated a probability distribution over these worlds by setting the values of the following fifteen independently variable probabilities: \(P(a), P(b|a), P(b|\neg a), P(c|a \land b), P(c|\neg a \land b), P(c|a \land \neg b), P(c|\neg a \land \neg b), P(d|a \land b \land c), P(d|a \land \neg b \land c), P(d|a \land b \land \neg c), P(d|\neg a \land b \land c), P(d|\neg a \land b \land \neg c), P(d|\neg a \land \neg b \land c), P(d|\neg a \land \neg b \land \neg c)\). Within [5], the probability distributions over the sixteen worlds were selected for each simulation, by setting the above fifteen conditional probabilities according to a uniform probability distribution on the unit interval. Diverging from [5], we controlled the entropy level of the probability distributions over the sixteen worlds. For each simulation, we chose a particular entropy level \(\delta\). Our program then proceeded by generating probability distributions in the manner of [5] until a distribution was generated whose entropy resided in the interval \([\delta-0.05, \delta+0.05]\).
To manage the search space in assessing the four LP systems, we restricted our attention to conditionals whose antecedent and consequent consist in conjunctions of literals. We also assumed that no propositional atom appears twice in any premise conditional or inferred conditional. These restrictions effectively limited the language under consideration to 464 conditionals (cf. [5]). We call the language composed of this set of 464 conditionals “L₄”. Drawing from L₄, we assumed that a small number of conditionals, so-called premise conditionals, together with their associated probabilities, were known to the reasoning systems. We further required that the probability associated with each premise conditional was at least 0.9. We chose the cut-off 0.9, since cases where the probability of the premise conditionals is relatively high represent a significant challenge for systems Z and QC (cf. [5]). In each simulation, the three premise conditionals were selected at random from among the sentences of L₄ whose probability was at least 0.9. We then allowed each LP reasoning system to infer, from the given premise conditionals, all of the conditionals, \( C \Rightarrow rD \), that follow according to the respective systems. For systems P, Z, and QC, the value \( r \), for each inferred conditional, was set to be one minus the sum of the improbabilities of the premise conditionals needed in deriving the conclusion. For system O, \( r \) was set to be the probability value of the least probable premise conditional needed for the derivation of \( C \Rightarrow D \) in O.

After determining which conclusions were inferred by the four systems, each system was assigned numeric scores for each of the conclusions that it inferred. The first scoring measure that we applied is called the advantage-compared-to-guessing measure. The idea behind this measure derives from the fact that the mean difference between two random choices of two real values \( r \) and \( s \) from the unit interval is (probably) \( 1/3 \). Based on this fact, we assessed each system by counting a judged lower probability bound that differs from the true probability by more than one-third negatively, and counting a judged lower probability bound that differs from the true probability by less than one-third positively. We scored the judged lower probability bounds by a simple linear measure of their distance from the true probabilities:

(6) The advantage-compared-to-guessing (ACG) score for derived conditionals:
\[
\text{Score}_{\text{ACG}}(C \Rightarrow rD, P) := \frac{1}{3} - |r - P(D|C)|.
\]

For reasons elaborated in [5], the ACG measure does not provide a fully adequate means of evaluating LP systems. In order to take a broad view of the advantages and disadvantages of reasoning in accordance with the four systems, we considered two other scoring measures.

We call the second measure that we considered the subtle-price-is-right measure. This measure assigns a positive score to any inferred lower probability bound that does not exceed the true probability, and penalizes inferred bounds that exceed the true probability by a simple linear measure of their distance above the true probability:
The subtle-price-is-right score for derived conditionals:

\[ \text{Score}_{\text{SPIR}}(C \Rightarrow r D, P) := r, \text{ if } r \leq P(D|C), \]
\[ := P(D|C) - r, \text{ otherwise.} \]

We call the final scoring measure that we considered the expected utility measure:

The expected utility score for derived conditionals:

\[ \text{Score}_{\text{EU}}(C \Rightarrow r D, P) := (P(D|C)^2 - (P(D|C) - r)^2) \cdot P(C)/2. \]

The EU measure scores an inferred conditional, \( C \Rightarrow D \), by evaluating the expected value of the decisions licensed by the acceptance of such a conditional (i.e., a conditional whose content is \( P(D|C) \geq r \)). In particular, we assume that a judged greatest lower conditional probability bound has the following behavioral import: If \( r \) is the greatest lower probability bound that a given agent accepts for \( D \) given \( C \), then (if she is prudent and has sufficient wealth) she will purchase all wagers on \( D \), conditional on \( C \), at price \( s \), so long as \( s < r \), and refuse to accept such wagers for \( s \geq r \). Given this behavioral interpretation of inferred conditionals, we considered an environment in which a respective agent is offered a single opportunity to purchase a wager on \( D \) conditional on \( C \) with a stake \( s \), where \( s \) is determined at random, according to a uniform probability distribution over the interval \([0, 1]\). In that environment, the expected value of accepting the greatest lower probability bound \( r \) on \( P(D|C) \) is provably:

\[ (P(D|C)^2 - (P(D|C) - r)^2) \cdot P(C)/2 \text{ (cf. [14]).} \]

3 The Results

The entropy of a probability distribution, \( P \), over a finite set of possible worlds, \( W \), is defined as \( E(P) = -\sum_{w \in W} P(w) \cdot \log(P(w)) \). So in the case where \( W \) contains sixteen worlds (as is the case in our simulations) \( E(P) \) will be in \([0, 4]\), where \( E(P) = 4 \) means that \( P \) is a uniform probability distribution over \( W \), and \( E(P) = 0 \) means that \( P \) is a standard valuation function (assigning the value 1 to exactly one world, and the value 0 to all others). Since the four LP systems that we consider are ordered in terms of the number of inferences they license (\( O \subset P \subset Z \subset QC \)), our focus here is on the 'new' inferences licensed by each system as one proceeds from system \( O \) to system \( QC \), i.e., the inferences licensed by system \( O \), the inferences licensed by system \( P \) that are not licensed by system \( O \) (\( P-O \)), the inferences licensed by system \( Z \) that are not licensed by system \( P \) (\( Z-P \)), and the inferences licensed by system \( QC \) that are not licensed by system \( Z \) (\( QC-Z \)). Table 1 lists the average number of conclusions inferred by each (sub)system, across varied entropy levels, and the average number erroneous inferences among the \( Z-P \) and \( QC-Z \) inferences, i.e., those instances

\(^5\) The name of the measure derives from the long running American game show where contestants must guess the price of items, and succeed by having the most accurate guess that does not exceed the price of the relevant item.
where the inferred lower bound exceeded the actual probability. (These average values are based on a sample of one thousand simulations at each listed entropy level.)

Table 1. Mean number of inferences and errors

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean Number of Inferences</th>
<th>Mean Number of Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
<td>P-O</td>
</tr>
<tr>
<td>3.5</td>
<td>3.02</td>
<td>0.06</td>
</tr>
<tr>
<td>3.0</td>
<td>3.1</td>
<td>0.25</td>
</tr>
<tr>
<td>2.5</td>
<td>3.28</td>
<td>0.4</td>
</tr>
<tr>
<td>2.0</td>
<td>3.56</td>
<td>0.76</td>
</tr>
<tr>
<td>1.5</td>
<td>4.01</td>
<td>1.23</td>
</tr>
<tr>
<td>1.0</td>
<td>4.3</td>
<td>1.84</td>
</tr>
<tr>
<td>0.5</td>
<td>4.9</td>
<td>2.82</td>
</tr>
</tbody>
</table>

The most obvious pattern exhibited in table 1 is that the number of inferences drawn by each system is a decreasing function of the entropy level. This pattern was expected, since lower entropy levels imply a less evenly distributed probability function, and in turn a greater number of possible premise conditionals with multiple conjuncts in their consequents. Such conditionals support a greater number of inferences in all of the systems considered.

We now consider the average scores earned by the respective systems for the full set of conclusions drawn within a single simulation. Tables 2, 3, and 4 list the results.

Table 2. Mean ACG scores

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean ACG Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
</tr>
<tr>
<td>3.5</td>
<td>1.01</td>
</tr>
<tr>
<td>3.0</td>
<td>1.03</td>
</tr>
<tr>
<td>2.5</td>
<td>1.08</td>
</tr>
<tr>
<td>2.0</td>
<td>1.17</td>
</tr>
<tr>
<td>1.5</td>
<td>1.31</td>
</tr>
<tr>
<td>1.0</td>
<td>1.40</td>
</tr>
<tr>
<td>0.5</td>
<td>1.600</td>
</tr>
</tbody>
</table>

Table 3. Mean sPIR scores

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean sPIR Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
</tr>
<tr>
<td>3.5</td>
<td>2.83</td>
</tr>
<tr>
<td>3.0</td>
<td>2.92</td>
</tr>
<tr>
<td>2.5</td>
<td>3.11</td>
</tr>
<tr>
<td>2.0</td>
<td>3.40</td>
</tr>
<tr>
<td>1.5</td>
<td>3.83</td>
</tr>
<tr>
<td>1.0</td>
<td>4.12</td>
</tr>
<tr>
<td>0.5</td>
<td>4.76</td>
</tr>
</tbody>
</table>
Examine the tables 2, 3, and 4, we see that the QC–Z inferences earn negative scores at every entropy level, according to all three scoring rules. This provides a relatively good reason for concluding that we should not reason in accordance with system QC, if our concern is to draw conclusions that are accurate and informative. On the other hand, we see that O and P–O inferences earn positive scores at every entropy level, according to all three scoring rules. So it pretty clear that it is reasonable to make these inferences. In fact, the present conclusion is unsurprising given (1) and (2), above, that characterize the ability of systems O and P to preserve premise probability.

It is only when we turn to evaluate the quality of Z–P inferences that the data from tables 1, 2, and 3 is equivocal. When considering the ACG and sPIR scores for the Z–P inferences, we observe a peak in performance, when the entropy level of the underlying probability distribution is relatively high (≈ 3.00), but thereafter decreasing entropy correlates with decreasing ACG and sPIR scores. On the other hand, decreasing entropy correlates with increasing EU scores. Figure 1 provides a graphical representation of that pattern.

---

**Table 4. EU scores**

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean EU Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
</tr>
<tr>
<td>3.5</td>
<td>0.232</td>
</tr>
<tr>
<td>3.0</td>
<td>0.462</td>
</tr>
<tr>
<td>2.5</td>
<td>0.656</td>
</tr>
<tr>
<td>2.0</td>
<td>0.867</td>
</tr>
<tr>
<td>1.5</td>
<td>1.191</td>
</tr>
<tr>
<td>1.0</td>
<td>1.520</td>
</tr>
<tr>
<td>0.5</td>
<td>1.975</td>
</tr>
</tbody>
</table>

---

**Fig. 1. Mean EU scores**
In order to get a clearer idea of what’s going on, it is helpful to look at the average scores earned for single inferences across varied entropy levels. Tables 5, 6, and 7 list the results, and figure 2 provides a graphic presentation of the information presented in Table 7.

**Table 5. Mean ACG scores per inference**

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean ACG Score per Inference</th>
<th>O</th>
<th>P-O</th>
<th>Z-P</th>
<th>QC-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td></td>
<td>0.333</td>
<td>0.275</td>
<td>0.234</td>
<td>-0.134</td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td>0.332</td>
<td>0.269</td>
<td>0.192</td>
<td>-0.173</td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td>0.331</td>
<td>0.275</td>
<td>0.130</td>
<td>-0.191</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td>0.329</td>
<td>0.273</td>
<td>0.075</td>
<td>-0.199</td>
</tr>
<tr>
<td>1.5</td>
<td></td>
<td>0.326</td>
<td>0.274</td>
<td>0.029</td>
<td>-0.202</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.325</td>
<td>0.278</td>
<td>-0.026</td>
<td>-0.207</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>0.327</td>
<td>0.302</td>
<td>-0.069</td>
<td>-0.221</td>
</tr>
</tbody>
</table>

**Table 6. Mean sPIR scores per inference**

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean sPIR Score per Inference</th>
<th>O</th>
<th>P-O</th>
<th>Z-P</th>
<th>QC-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td></td>
<td>0.939</td>
<td>0.863</td>
<td>0.098</td>
<td>-0.447</td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td>0.944</td>
<td>0.866</td>
<td>0.160</td>
<td>-0.471</td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td>0.950</td>
<td>0.878</td>
<td>0.101</td>
<td>-0.489</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td>0.953</td>
<td>0.887</td>
<td>0.029</td>
<td>-0.492</td>
</tr>
<tr>
<td>1.5</td>
<td></td>
<td>0.954</td>
<td>0.898</td>
<td>-0.035</td>
<td>-0.489</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.958</td>
<td>0.906</td>
<td>-0.114</td>
<td>-0.493</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>0.972</td>
<td>0.949</td>
<td>-0.176</td>
<td>-0.515</td>
</tr>
</tbody>
</table>

**Table 7. Mean EU scores per inference**

<table>
<thead>
<tr>
<th>Entropy Level</th>
<th>Mean EU Score per Inference</th>
<th>O</th>
<th>P-O</th>
<th>Z-P</th>
<th>QC-Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td></td>
<td>0.077</td>
<td>0.086</td>
<td>0.038</td>
<td>-0.00015</td>
</tr>
<tr>
<td>3.0</td>
<td></td>
<td>0.149</td>
<td>0.165</td>
<td>0.055</td>
<td>-0.00124</td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td>0.200</td>
<td>0.222</td>
<td>0.063</td>
<td>-0.00133</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td>0.243</td>
<td>0.283</td>
<td>0.067</td>
<td>-0.00137</td>
</tr>
<tr>
<td>1.5</td>
<td></td>
<td>0.297</td>
<td>0.344</td>
<td>0.075</td>
<td>-0.00137</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.354</td>
<td>0.404</td>
<td>0.080</td>
<td>-0.00138</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>0.403</td>
<td>0.458</td>
<td>0.084</td>
<td>-0.00103</td>
</tr>
</tbody>
</table>
Our main remaining concern is to evaluate the quality of $Z-P$ inferences. Tables 5 show that $Z-P$ inferences lead to bounds that are relatively close to the true probabilities, when entropy high. However, when the entropy level is very low, the distance between the judged bounds and the true probability tends to be rather great. For example, when the entropy of the underlying distribution is 0.5, the inferred lower bound for an average $Z-P$ inference differs from the true probability by about 0.4. Similarly, while $Z-P$ inferences are expected to yield relatively high sPIR scores, when an inferred bound is not in error (ranging from about 23 to 38 percent of cases, depending on the entropy level), we see that when the entropy level is low, a typical erroneous inferred bound exceeds the true probability by a significant margin. In contrast, the EU scores for $Z-P$ inferences (as with $O$ and $P-O$ inferences) increases with decreases in the entropy of the underlying probability distribution. The latter result marks one positive sign in favor of the quality of $Z-P$ inferences. And we maintain that the latter result does reflect a significant capacity of $Z-P$ inferences to exploit information about an environment to draw helpful conclusions about that environment. Indeed, if we consider plausible aprioristic methods of assigning lower probability bounds, such as the ones considered in [14], i.e., methods of assigning lower probability bounds to the elements of $L_4$ without exploiting the information that was supplied to the four LP systems (in the form of premise conditionals), then we see that the EU scores earned for $Z-P$ inferences tend to be much higher than the scores earned by aprioristic methods. For example, the most successful aprioristic method considered in [14] assigned the lower bounds 1/2, 1/4, and 1/8, respectively, to conditionals with one, two, or three conjuncts in their consequent (as the values 1/2, 1/4, and 1/8 are the average probabilities for conditionals with the corresponding number of conjuncts in their consequents). In the case where entropy was not con-

---

6 The present effect is the result of inferred conclusions with more probable antecedents, when the entropy level is low.
trolled (and the mean entropy of the underlying probability distributions was about 2.88), this aprioristic method earned an EU score of about 0.0204 per inference, which is far lower than the average scores earned by $Z-P$ inferences (across all entropy levels).

4 Conclusions

It almost goes without saying that it is reasonable to accept the conclusions of $O$ and $O-P$ inferences, so long as our goal is to accept accurate and informative probability statements. It is also quite clear that we should not accept the conclusions of $QC-Z$ inferences. The difficult choice is whether to accept the conclusions of $Z-P$ inferences. In cases where it is known that the entropy of the underlying distribution is not low (or probably not low), it will usually be reasonable to shoulder the risk inherent in accepting the conclusions of $Z-P$ inferences. More generally, the tendency of $Z-P$ inferences to deliver significant positive EU scores (even when the entropy of the underlying distribution is very low) indicates the value of these inferences as a basis for decision making.

In considering whether it is reasonable to accept the conclusions of $Z-P$ inferences, we think it is reasonable to consider whether there are alternatives that would support better probability judgments. Since we know that our present method of associating lower probability bounds with $Z-P$ inferences is prone to overestimation, we conjecture that a more optimal method would make a downward correction to these assigned bounds. It would also make sense to vary the size of this correction, in cases where the entropy of the underlying distribution is known. The exploration of this idea is left to future work.

Acknowledgments. Work on this paper was supported by the DFG-Grant SCHU1566/5-1 as part of EuroCores LogiCCC project The Logic of Causal and Probabilistic Reasoning in Uncertain Environments (LcpR), and the DFG-Grant SCHU1566/9-1 as part of the SPP 1516 priority program “New Frameworks of Rationality”.

References

Quantitative Measures for Adaptive Object-Oriented World Modeling

Achim Kuwertz\textsuperscript{1} and Jürgen Beyerer\textsuperscript{1,2}

\textsuperscript{1} Vision and Fusion Laboratory (IES), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT) - Karlsruhe, Germany
achim.kuwertz@kit.de

\textsuperscript{2} Fraunhofer IOSB - Institute of Optronics, System Technologies and Image Exploitation - Karlsruhe, Germany
juergen.beyerer@iosb.fraunhofer.de

Abstract. Adaptive knowledge modeling is an approach to extend the abilities of the Object-Oriented World Model, a system for environment representation, for allowing it to cope with open environments in which unforeseen entities can occur. In previous work, adaptive knowledge management was introduced and a quantitative measure rating the quality of domain models was presented. In this contribution, the approach is extended and further measures, designed for identifying points of necessary change in knowledge models, are proposed. In addition, an approach for adapting the knowledge model based on the identified points of change is presented.

1 Introduction

Representing the current state of an environment is a task that often arises when dealing with artificial cognitive systems or systems supporting situation assessment. A world modeling system can support this task by providing a structured way for organizing and managing the environment information that is acquired by sensing systems. The Object-Oriented World Model (OOWM, [1]) is a world modeling systems that has been successfully applied to different domains, including autonomous service robotics and maritime situation assessment. The OOWM is designed as a probabilistic architecture for processing observations of real-world entities, provided by heterogeneous sensing systems, and for creating an internal representation that depicts the state of an observed environment. For classifying observed entities, an a priori generated domain model is used as background knowledge. Based on this domain model, semantic interaction and reasoning is enabled. Since the model is generated a priori, it is only able to represent a finite and closed part of an application domain.

During operations of the OOWM, it may occur that entities are observed which are not represented in the domain model, and thus cannot be properly processed by the OOWM. In order to cope with such situations, an adaptive approach for managing the knowledge model of the OOWM is needed. Such an
approach has been proposed in [2]. The overall approach comprises methods for evaluating the quality of the knowledge model with regard to its ability for representing the currently observed world, as well as methods for performing model adaptations in case of decaying quality. As a prerequisite for model adaptation, points of necessary change have to be identified. In [2], a measure for rating the quality of a knowledge model based on the Minimum Description Length principle [3] has been proposed. This contribution now focuses on identifying points of change, i.e., entities that are not well represented within the given knowledge model, as well as on approaches for adapting the knowledge model. Related work to adaptive knowledge modeling concerns the areas of unsupervised classification and concept acquisition, like Numerical Taxonomy [4, 5] or Conceptual Clustering [6, 7], and is given in more detail in [2].

In Sec. 2, an overview of the OOWM is given, including a new association approach. Adaptive knowledge modeling is presented in Sec. 3. Measures for identifying points of model change and adaptation approaches are proposed in Sec. 4. In Sec. 5, a small example scenario is provided as a proof of concept.

2 World Modeling for Cognitive Systems

One key capability of a cognitive system is the ability to manage the information it has observed about the environment it operates in. This includes consolidating and integrating such information, persistently storing it, as well as making it available to respective processing modules. Acting as a memory structure and information hub, the OOWM [1, 8] can be employed to fulfill such tasks. Being designed as a probabilistic framework for data and information fusion, the OOWM can handle environment observations provided by heterogeneous sensing systems and, by consistently integrating the contained information, is able to represent the current and historic state of an observed environment.

2.1 The Object-Oriented World Model

The OOWM is a general framework for probabilistic world modeling. It is composed of two components, as illustrated in Fig. 1, which are responsible for handling observation data on the one hand and a priori domain knowledge on the other hand. In its general functioning, the OOWM is employed to process observations of entities from a spatio-temporal section of the real world, the application domain, which can be provided by various sensing systems. Such observations carry information about real-world entities like their position, size or color, which in consequence get stored in the dynamic modeling part of the OOWM, denoted as World Model.

In the World Model, observed entities are represented as information objects called representatives. A representative \( R \) can be seen as the set of observed entity attributes \( A_R := \{A_1, A_2, \ldots, A_n\}, n \in \mathbb{N} \), with \( A_i \) e.g. representing the color of an observed real-world entity. Observed entity attributes are
described by probability distributions $p_{A_i}(a)$, interpreted as the degree of belief (DoB) in the observed value, and are assumed to be stochastic independent. Representatives can thus be described by joint probability distributions as $p(R) = p(A_R) = \prod_{i=1,...,n} p_{A_i}(a)$. For describing attribute distribution, either discrete probability distributions $p_{A_i}^d$ with domain $\mathcal{S}_A$ or continuous distribution $p_{A_i}^c$ (especially Gaussian and Gaussian mixture distributions) are employed to represent nominally or ordinally scaled attributes on the one hand and cardinally scaled attributes on the other hand.

Fig. 1. Structure and basic functioning of Object-Oriented World Modeling. Features of entities in a considered real-world domain are observed by sensing systems and get stored in the World Model by creating representatives. In Background Knowledge, a domain model is used to capture the semantics of the considered domain by defining interrelated concepts. Representatives can get associated to these concepts for enabling a meaningful information processing.

The second component of the OOWM is the Background Knowledge, in which knowledge about the application domain is stored as the result of an a priori conceptualization process. The resulting knowledge model contains object-oriented class-like descriptions of all relevant real-world entities of the application domain, denoted as concepts. A concept $C$ is characterized by its set $\mathcal{A}_C := \{A_1, A_2, \ldots, A_m\}$ of modeled entity attributes and a number of relations necessary for the modeled entity (e.g., part of relations). Modeled entity attributes are described by DoB probability distributions $p_{A_i}(a)$, and concepts are represented as the joint probability distribution $p(C) = p(A_C) = \prod_{c=1,...,m} p_{A_i}(a)$ over their attribute distributions. The Background Knowledge can be employed to enrich the information available about observed entities: by associating a representative to a defined concept based on the observed attributes, additional attributes modeled in the concept can be derived from the knowledge model, e.g., attributes that have not been observed yet or information on unobservable features like the function or role of an entity.

Information processing in the OOWM takes place mainly at the World Model. This processing is performed probabilistically and, in general, follows the Bayesian methodology. New observations received from sensing systems are associated to
existing representatives using probabilistic data association methods based on location information. If no appropriate representative is contained in the World Model, a new one gets created. If associated to an existing representative, a new observation triggers an update on this representative by either extending its set $A_R$ of observed attributes with the new attribute observation $A_{n+1}$, or by updating the DoB distribution of an existing attribute $A_i \in A_R$. Updates of DoB distributions are performed based on a Kalman filter and derivatives approach. More details on the general OOWM approach can be found in [1].

2.2 Representative-to-Concept Association

The two components of the OOWM, the dynamic World Model and the a priori Background Knowledge, are connected by a probabilistic classification mechanism, which tries to associate representatives $R$ to concepts $C$ by means of the conditional probability distribution $p(C|R)$, denoted as association probability. The association probability needs to be evaluated every time new information about observed entities, i.e., a new observation, is available. In the case of an adaptive knowledge model, it additionally needs to be updated when changes in the definitions of concepts occur. The association probability is intended to rate, for a representative $R$ and a concept $C$, how well the DoB distributions of the observed attributes $A_R$ correspond to the distributions of the modeled attributes $A_C$ defined for the concept $C$. It itself constitutes a discrete distribution over all concepts $C$ in the set $C$. The association of representatives to concepts is a kind of multiclass categorization problem, but, unlike in most classification approaches, a probability distribution is produced as result (as e.g. is the case for Bayesian classifiers). Besides classification, this distribution can be of use in further tasks, e.g., for model evaluation as described in Sec. 3.2.

A New Approach to Association In the OOWM, the association probability can be calculated as described in [9] for a hierarchically organized knowledge model using rejection classes. This approach requires a rather complex processing and does not seamlessly integrate into probabilistic processing. In this contribution, an alternate approach for calculating the association probability is presented, which aims at providing a simple and analytical solution being fully integrated into Bayesian estimation theory. The approach as presented is hierarchy agnostic. The calculation of the association probability in the proposed approach is based on computing conditional probability distributions $p(C|A_i)$ over the concepts $C \in C$, conditioned on the observed DoB distribution $p_{A_i}(a)$ for each attribute $A_i \in A_R$ of a representative. These attribute-based association probabilities describe the probability that the representative $R$ should be associated to a concept $C$ when only considering observed values for one attribute. In order to calculate the representative-to-concept association probability $p(C|R) = p(C|A_1, A_2, \ldots, A_n)$ based on all attributes of the representative, the attribute-based probability distributions have to be adequately combined. When comparing an attribute distribution $p_{A_i}(a)$ to a modeled concept distribution $p_{A_i}(a)$ for computing the attribute-based association probability $p(C|A_i)$,
a measure for rating, descriptively speaking, how much the observed attribute
distribution is contain or overlapped by the concept distribution \( p_{A_i}(a) \) shall be
employed. Such a measure can for example be realized by calculating the inte-
gral \( \int p_{A_i}(a) \cdot p_{A_i}(a) \, da \) over the product of observed and modeled distributions.
Keeping this in mind, a more strict derivation of how to compute the association
probability will follow.

**Estimating the Association Probability** For associating representatives to
corcepts by the means of a probability distribution, estimation approaches, as
described e.g. in [10], are well suited. The goal then is to estimate the proba-
bility distribution \( p(c|r) \) of a discrete random variable \( c \) given an observation \( r \)
and a probabilistic mapping of \( c \) to \( r \) described by the conditional probability
distribution \( p(r|c) \). Given an observation value for \( \hat{r} \), backward inference can be
applied for calculating \( p(c|\hat{r}) \) according to Bayes’ theorem as

\[
p(c|\hat{r}) = \frac{1}{z} \cdot p(\hat{r}|c) \cdot p(c),
\]

(1)

where \( z \) is a normalization constant and \( p(c) \) describes the a priori distribution
of \( c \). In the OOWM, (1) can be employed to calculate the attribute-based asso-
ciation probability \( p(C|A_i) \) by interpreting \( c \) as a random variable over \( C \) and \( r \)
as the observed distribution of an attribute \( A_i \). A problem that arises with this
interpretation is that, in the OOWM, no observation value \( \hat{r} \) is given, but an
observed attribute distribution \( p_{A_i}(a) \). In consequence, one has to calculate a
probability distribution conditioned on another distribution, which is not possi-
bile in standard probability theory [10]. To circumvent this obstacle, an approach
described in [10] can be employed. In this approach, an additional mapping to a
dummy random variable \( d \) is introduced in order to be able to state the problem
in sound way. This results in a mapping from \( c \) to \( r \) to \( d \), each described by a
conditional probability distribution. The goal now is to estimate to distribution
\( p(c|\hat{d}) \), where the value of \( \hat{d} \) is calculated based on a given desired distribution
\( p(r) \) for \( r \). Following [10], the result \( p(c|\hat{d}) \) then can be calculated according to

\[
p(c|\hat{d}) = \frac{1}{z} \cdot p(c) \int_R p(r) \cdot p(r|c) \, dr,
\]

(2)

where \( z \) again is a normalization constant and \( p(c) \) is the prior distribution of
\( c \). Applying (2) to the calculation of the attribute-based association probability
results in

\[
p(C|A_i) = \frac{1}{z} \cdot p(C) \int_R p_{A_i}(a) \cdot p_{A_i}(a) \, da,
\]

(3)

where the observed attribute distribution \( p_{A_i}(a) \) takes the role of the given
desired distribution \( p(r) \), and the concept-dependent modeled attribute distri-
bution \( p_{A_i}(a) \) takes the role of the probabilistic mapping \( p(r|c) \). As prior distri-
bution \( p(C) \), a uniform distribution can be assumed.

As can be seen, (3) matches the intuition described earlier about how a
measure for an attribute-based association should be constructed. For discrete
attribute distributions \( p^d_A(a) \), the integral can be replaced by a sum. For calculating the representative-to-concept association \( p(C|R) = p(C|A_1, A_2, \ldots, A_n) \), the computed attribute-based association probabilities have to be combined. Based on the assumption of stochastic independence of attribute distributions, an iterative approach can be taken: given \( p(C|A_1, A_2, \ldots, A_{k-1}) \) with \( k-1 < n \), the distribution \( (C|A_1, A_2, \ldots, A_k) \) can be computed according to (3) by employing \( p(C|A_1, A_2, \ldots, A_{k-1}) \) as the prior distribution of concepts \( p(C) \).

In conclusion, the representative-to-concept association probability \( p(C|R) \) can be calculated according to

\[
p(C|R) = \frac{1}{z} \cdot p(C) \cdot \prod_{A_i \in A_R} \left( \int_{\mathbb{R}} p_{A_i}(a) \cdot p_{A_i}(a) \, da \right),
\]

(4)

with \( A_i \in A_C \) being the concept attribute corresponding to \( A_i \) and \( p(C) \) being an initial prior distribution. This allows to describe the calculation of the association probability in a simple closed form.

As a proof of concept, Fig. 2 displays some association results based on (4). The set of concepts was given as \( C = \{\text{Apple, Banana, Coconut}\} \), and the representatives Apple1, Coconut1 and Pear1 were associated, based on the definitions for fruit concepts and entities given in Sec. 5. As can be seen, the proposed association approach is able to associate the Apple1 representative to the Apple concept as well as the Coconut1 to the Coconut concept. For the Pear1 representative, the association probability is inconclusive as expected, since no matching concept is contained in \( C \).

![Fig. 2. Exemplary association probabilities for different representatives and the set of concepts \( C = \{\text{Apple, Banana, Coconut}\} \), as produced by the proposed new approach for representative-to-concept association.](image)

### 3 Adaptive Knowledge Modeling

The domain knowledge encoded into OOWM Background Knowledge a priori by human experts represents all the concepts that are considered relevant for the operations of the OOWM at design-time. Such an a priori model is only able to represent a closed set of concepts from the application domain. In many
cases, this concept set may be sufficient to support OOWM operations at run-
time. However, cognitive systems often have to deal with real-world scenarios
in dynamic domains which involve human interaction. In such scenarios, there
is a significant possibility to encounter real-world entities that have not been
considered in the domain model. Therefore, a world modeling system acting
as the central information hub for a cognitive system should be able to cope
with situations in which unforeseen entities are encountered. In consequence, an
approach for extending the knowledge model in order to support an adaptive
modeling of an open world is required. In the OOWM, this corresponds to man-
aging the set of concepts in Background Knowledge in adaption to the observed
representatives in the World Model.

3.1 The Adaptive Knowledge Modeling Approach

The adaptive management of a knowledge model is a complex task that subsumes
several different, but interrelated aspects. One important aspect is model eval-
uation, i.e., the set of all tasks related to evaluating the quality of a knowledge
model. Model evaluation comprises of global evaluation tasks, which rate the
overall quality of the entire knowledge model with respect to how well a model
is able to represent the observed real world, on the one hand, and, on the other
hand, how complex a model is. Based on such measures, a degradation in overall
model quality can be detected. In addition to these global tasks, there are local
evaluation tasks operating on only small subsets of the represented entities or
concepts, like the detection and identification of poorly described representatives
or the detection of similar concepts.

Besides model evaluation, model adaptation is the second important aspect
of adaptive knowledge model management. If, based on the evaluation of global
or local model quality measures, the need for changing the knowledge model is
detected, model adaptation tasks have to cope with assessing different alterna-
tives for model change (e.g., adjusting concepts vs. creating new ones) as well
as the actual adaptation of the knowledge model (e.g., by extending existing
concept descriptions or learning new ones). Model adaptation tasks are always
aimed at (re-)improving global model quality.

3.2 Quantitative Evaluation of Overall Model Quality

The evaluation of overall model quality, i.e., a quantitative rating of how well the
OOWM knowledge model is able to represent observed entities, is the central
part of adaptive knowledge management. In [2], we presented a framework for
the quantitative evaluation of overall knowledge model quality in the OOWM.
The approach is based on rating model quality using the Minimum Description
Length (MDL) principle (e.g., [3]). MDL as an information criterion in inductive
inference is aimed at selecting models which allow to describe observed data with
a minimal description length. The basic idea of MDL is to extract regularities
from observation data and use these regularities as a model to compactly describe
the data by only encoding their deviation from the model, and, in addition, the
extracted model itself. The better the model, the less encoding length is thus required to represent observed data. Therefore, MDL selects those models which results in a lower overall description length.

**Crude MDL** A simple approach to this idea is realized in crude MDL [3]. The overall quality of a model \( M \) is rated by a measure for the combined description length \( L(M) \) given as the two-part term \( L(M) = L(D|M) + L(M) \). The two parts represent the length \( L(D|M) \) of a description needed to represent the data \( D \) based on employing the model \( M \) for describing the data, as well as the term \( L(M) \) which rates the description length for the model itself and penalizes overly complex models (e.g., in case of overfitting). This basic approach for rating model quality is applied in OOWM model quality evaluation. In the OOWM, the parameters relevant for quality rating is the knowledge model itself, represented as the set of concepts \( C \) defined on basis of their attributes \( A_C \), the set of representatives \( R \) subsuming the observed attributes \( A_R \), as well as the association probability \( p(C|R) \) of representatives \( R \in R \) to concepts \( C \in C \). Based on this formalization, a quantitative measure for model quality \( Q(\cdot) \) following the principles of crude MDL can be defined as

\[
Q(R, C) = L(R|C) + L(C). \tag{5}
\]

The two-part approach rates the level of correspondence of observed data to modeled concepts by the term \( L(R|C) \) as well as the complexity of the knowledge model by the term \( L(C) \).

**Model Complexity** For rating model complexity, in [2] we proposed a measure which rates the overall complexity of the knowledge model \( L(C) \) as the sum of concept complexities \( L(C), C \in C \). The complexity of a concept is rated based on how specific its attributes are, i.e., as the sum of the specificity \( L(p_{A_c}(a)) \) of the DoB distributions \( p_{A_c}(a) \) representing the attributes \( A_c \) of a concept \( C \). Thus, the cumulative measure

\[
L(C) = \sum_{C \in C} \sum_{A_c \in A_C} L(p_{A_c}(a))
\]

results. For rating the specificity of DoB distributions, concentration measures based on Shannon entropy \( H(\cdot) \) are employed, e.g., for rating the specificity of a discrete attribute \( A^d \) with support \( S_A \) according to

\[
L(A^d) = \log(|S_A|) - H(A^d). \tag{6}
\]

Continuous attributes are handled by discretizing them first according to an Least Discernible Quantum approach (see [2] for details).

**Model Correspondence** The second term in the model quality measure \( Q(R, C) \), rating the correspondence \( L(R|C) \) of so far observed entities to modeled concepts,
can also be calculated cumulatively as the sum

\[ L(R|C) = \sum_{R \in R} \sum_{C \in C} p(C|R) \cdot L(R|C) \] (7)

over all representatives \( R \) being currently stored in the World Model. The concept correspondence \( L(R|C) \) is designed to rate how well a given representative \( R \) matches a concept \( C \), thereby being a measure of how much description is necessary to encode \( R \) given \( C \) as its model. The matching is calculated based on the corresponding attributes \( A_{cr} \) of \( C \) and \( A_i \) of the representative \( R \) according to

\[ L(R|C) = \sum_{A_i \in A_R} w_{A_{cr}} \cdot L(A_i|A_{cr}) \] (8)

allowing for an individual attribute weights \( w_{A_{cr}} \). The attribute correspondence \( L(A_i|A_{cr}) \) thereby constitutes a function for comparing probability distributions, e.g., an information-theoretic measure like the Kullback Leibler divergence \( D_{KL}(A_i||A_{cr}) = \sum_{a \in A_i} p^d_{A_i}(a) \cdot \log(p^d_{A_i}(a) / p^d_{A_{cr}}(a)) \) or cross entropy (e.g., [11]). Furthermore, a discrete scaled overlap measure \( L_o(A_i, A_{cr}) = \sum_{a \in A_i} l_o(a) \leq 1 \) with

\[ l_o(a) := \begin{cases} 0, & \text{if } \tilde{p}^d_{A_i}(a) \leq p^d_{A_{cr}}(a) \\ \left| p^d_{A_i}(a) - p^d_{A_{cr}}(a) \right| & \text{otherwise} \end{cases} \] (9)

is employed with a scaled variant \( \tilde{p}^d_{A_i} \) of \( p^d_{A_i} \). More details on correspondence measures is given in [2].

### 3.3 Towards Model Adaptation

In order to improve the quality of a knowledge model by adaptation, starting points for improvement have to be identified first. In case of the OOWM Background Knowledge, the representatives that are only poorly mapped by the current knowledge model, i.e., not well described by any of the concepts contained in \( C \), are a promising starting point for model adaptation. These representatives are denoted as *poorly mapped representatives*. So, as a prerequisite for model adaptation, such representatives have to be detected and identified. How this can be done is presented in detail in the following chapter.

### 4 New Contributions to Adaptive Knowledge Modeling

Adaptive management of knowledge models is aimed at maintaining the quality of a knowledge model with regard to how well the model is suited for the desired tasks a cognitive systems is to perform. For the OOWM, the most important aspect is the interoperable and semantic representation of a considered application domain. Complementing the framework for quantitative model evaluation summarized in Sec. 3, this section presents ideas on different approaches for detecting and identifying poorly mapped representatives (PMR) as the potential starting points for model adaptation as well as on approaches for performing such adaptation based on the identified PMRs.
4.1 Detection and Identification of Poorly Mapped Representatives

For detecting and identifying representatives that are only poorly mapped in the current Background Knowledge, local and global approaches can be distinguished. Local approaches are based only on the current set of representatives $R$ in the World Model and the current set of modeled concepts $C$ in Background Knowledge or respective measures on these sets, like the current association probabilities $p(C|R)$, and treat each representative individually. Global approaches, however, consider the complete history of the World Model, i.e., the sets of representatives $R_t$ for all the points of time $t > 0$ the OOWM has been operating, and are based on measures accumulating over representatives, and/or concepts, like the overall model quality (5). Global measures can also be cumulative over different points of time.

In general, local approaches are well-suited for identifying points of change in knowledge models (e.g., PMRs), whereas global measures are advantageous for detecting the need for model adaptation on a strategic level (e.g., decaying quality) as well as guiding the direction of this adaptation. In the following, several local and global approaches are introduced. A initial evaluation of their suitability is given within the example scenario presented in Sec. 5.

Rejection Classes The simplest case of detecting and identifying PMRs, seen from the perspective of a knowledge management mechanism, is given when the employed association mechanism is equipped with rejection classes. These classes can be used to tag representatives that cannot be associated to any concept in the knowledge model. Tagged representatives can then be passed directly to a model adaptation mechanism. For the association approach (4) presented in Sec. 2.2, a similar behavior can be achieved by directly considering the unnormalized probabilities for the representative-to-concept association and detecting those cases where only low association probabilities are encountered.

Association-based Measures The first proposed measure for identifying PMRs is a local measure based on the representative-to-concept association probabilities $p(C|R)$, as provided by an association mechanism. The approach to identifying PMRs using these probabilities rests on the following assumptions. If a representative $R$ is well described by any concept $C$, this concept will be assigned a high probability in $p(C|R)$. If none of the concepts is suited to describe the representative well, the association will be inconclusive and the association probability will resemble a kind of noisy uniform distribution. The latter case is the event that we want to detect, since it symbolizes a PMR. So, the critical factor for distinguishing between well and poorly mapped representatives, based on a given association, is the way how the mass of the association probability is concentrated. In order to distinguish different concentrations, a measure based on Shannon entropy $H(\cdot)$ can be employed for identifying PMRs. This approach is similar to the rating of attribute specificity in Sec. 3.2.

For assessing this approach, different measures based on the difference of the current entropy $H(p(C|R))$ of the association probability to the maximum
entropy of any discrete distribution over the respective value range (i.e., \( \log(|C|) \), representing the most inconclusive association) have been evaluated. The most promising measure was found to be given by the relative entropy difference

\[
M(p(C|R)) := \frac{\log(|C|) - H(p(C|R))}{H(p(C|R))}.
\] (10)

**Fig. 3.** Exemplary values of the association-based measure \( M(p(C|R)) \) used to identify poorly mapped representatives (PMRs). In the first row, values as well as association probabilities are depicted which symbolize PMRs. The second row shows well described representatives for different sizes of \( C \).

Figure 3 shows some exemplary values of \( M(\cdot) \) for different association probabilities \( p(C|R) \) and different sizes of \( C \). In the first row, values symbolizing PMRs are shown. The second row shows values for well described representatives. As can be seen, a decision threshold of \( \phi_M \approx 0.5 \) could be selected in order to distinguish poorly from well mapped representatives. In addition, the association-based measure \( M(\cdot) \) allows to conclude whether concepts exist in the knowledge model that are, in some way, similar to the examined PMR. This information can then be used as a hint for a model adaptation mechanism for not creating a new concept but extending an existing one. In the given example, e.g., values within the interval \([0.6, 1]\) could be seen as such indications.

**Correspondence-based Measures** As alternate local measure for PMR identification, the concept correspondence (8), as defined in Sec. 3.2, can be employed. This measure computes a correspondence value for a given representative \( R \) to a concept \( C \) in Background Knowledge. The higher the value, the less likely the correspondence to this concept is. A PMR can thus be characterized
by having high correspondence values to all the concepts in $C$. To illustrate the correspondence-based approach, Fig. 4 displays some example correspondence values for the representatives Pear 1, Strawberry 1, Apple 1 and Banana 1 to the set of concepts $C = \{\text{Apple, Banana, Coconut}\}$ - the used representatives and concepts are defined as described in Sec. 5. As can be seen, it is again possible to determine a decision threshold, $\phi_L$, for this measure in order to identify PMRs - e.g., $\phi_L \approx 2.5$.

![Figure 4](image-url)

**Fig. 4.** Example of correspondence values based on (8) for different representatives to the set of concepts $C = \{\text{Apple, Banana, Coconut}\}$. These values can be used in a correspondence-based measure for identifying PMRs.

**Global Measures** Global measures for the detection and identification of PMRs are used over a series of representatives and take into account the history of the World Model, i.e., the evolution of the set of representatives over time. Based on the model correspondence (7), a global measure for detecting PMRs at the point of time $t = t_x$ can be defined according to

$$Q_{C}(\mathcal{R}, C) = g\left( L(R_{t_0} | C), L(R_{t_1} | C), \ldots, L(R_{t_x} | C) \right),$$

where $g(\cdot)$ is a function over the model correspondence values $L(R_t | C)$ at different points of time. By using different evaluation functions $g(\cdot)$, different global measures can be defined. The absolute step measure $Q_{C,1}^a(\mathcal{R}, C)$ for example employs an evaluation function $g_{1}^a(\cdot) = L(R_{t_x} | C) - L(R_{t_{x-1}} | C)$ for rating the absolute increase over a single time step for PMR detection. The respective decision threshold can be derived on the basis of $\phi_L$. Another example is the relative step measure $Q_{C,1}^r(\mathcal{R}, C)$ which employs a moving average of the stepwise increase, described by the evaluation function

$$g_{1}^r(\cdot) = \left( L(R_{t_x} | C) - L(R_{t_{x-1}} | C) \right) - \sum_{x=1}^{x-1} \left( \frac{L(R_{t_x} | C) - L(R_{t_{x-1}} | C)}{x-1} \right).$$

**Definition of Decision Thresholds** In this section, several local and global measures for detecting and identifying PMRs have been proposed. Those measures employ thresholds $\phi$ for deciding whether a given representative is poorly represented. As a proof of concept, examples were shown how such decision thresholds can be defined. For applying these measures within a system for
adaptive knowledge management, further analyzes of their properties are necessary, including for example how values for the employed thresholds $\phi$ could be determined by automated procedures, e.g., machine learning approaches, based on the given data and employed knowledge models.

### 4.2 Approaches to Model Adaptation

Model adaptation is responsible for changing the knowledge model in a way that re-improves its quality, i.e., its ability to represent the observed application domain. As mentioned in Sec. 3.1, model adaptation can, for example, change a knowledge model by adding newly defined concepts or by extending existing concept definitions. In the following, ideas on how such model adaptation could proceed are presented. As starting point for model adaptation, the PMRs identified by the methods presented in Sec. 4.1 can be used.

When performing model adaptation, several different situations can be distinguished. The simplest case is given when only one representative $\bar{R}$ is supplied as a PMR, for which a new concept definition is to be created. In this case, a new concept definition can be created by simply using the observed attributes $A_{\bar{R}}$ of $\bar{R}$ as the modeled attributes $A_C$ for the newly defined concept $C_{\bar{R}}$.

A more complex situation is given when a number of PMRs are supplied which all have been identified as instances of one unknown concept to be newly created. In this case, the attributes $A_i$ of the PMRs have to be combined individually. The resulting attribute set can then be used as the set of modeled attributes for the new concept to be defined. Given the set $\mathcal{R}_P$ of PMRs, for each discrete attribute $A^d_i$ of a representative $\bar{R} \in \mathcal{R}_P$ a combined attribute $A^d_i$ can be calculated as the mean $\left( \sum p^d_{A_i}(a) / |\mathcal{R}_P| \right)$ over the distributions $p^d_{A_i}(a)$ for all given PMRs. For handling the continuous attributes $A^c_i$ of representatives in $\mathcal{R}_P$, a Gaussian mixture distribution can be constructed. In order to generalize beyond a pure sum of component distributions, mixture reduction approaches can be employed. In [12], a suitable approach for reducing the mixture to one single Gaussian distribution is described. If the resulting distribution shall fulfill quantitative requirements, the approach described in [13] is better suited.

Another possible situation is the extension of an existing concept by a given PMR. This case can be mapped to the multi-PMR case described above, by performing a weighted combination of the original concept attribute and the respective PMR attribute for each attribute of the extended concept. The original concept attributes can e.g. be weighted by a factor correlating to the number of representatives that have originally been used for defining the concept.

For implementing the proposed approaches for model adaptation, further elaboration and refinement is needed as future work. Furthermore, their interplay with model evaluation and their effectiveness for model improvement have to be evaluated.
5 Evaluation and Proof of Concept

For demonstrating the interplay of the presented association approach (4), the detection measures (e.g., (7), (10)) and the model evaluation framework, and as an initial proof of concept, a small example scenario for modeling a kitchen scene was considered. In this scenario, different fruits are considered as entities of interest. In Table 1, the set of concepts $C = \{\text{Apple, Banana, Coconut}\}$, used as Background Knowledge, is given. Concepts are defined using a discrete color attribute and a continuous length attribute (given in $m$). As sample representatives, two apples, a banana, a coconut, a pear, and a strawberry are used, defined according to Table 2.

<table>
<thead>
<tr>
<th>Apple1</th>
<th>Apple2</th>
<th>Coconut1</th>
<th>Pear1</th>
<th>Strawberry1</th>
<th>Banana1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{N}(8, 0.5)$</td>
<td>$\mathcal{N}(7, 0.2)$</td>
<td>$\mathcal{N}(15, 0.75)$</td>
<td>$\mathcal{N}(13, 0.2)$</td>
<td>$\mathcal{N}(3.5, 1)$</td>
<td>$\mathcal{N}(17, 0.1)$</td>
</tr>
<tr>
<td>green-yellow</td>
<td>red</td>
<td>brown</td>
<td>green-yellow (3:1)</td>
<td>red</td>
<td>yellow</td>
</tr>
</tbody>
</table>

Table 1. The set of concepts $C = \{\text{Apple, Banana, Coconut}\}$ in the exemplary kitchen scenario, defined by a discrete color attribute and a continuous length attribute (in $m$).

<table>
<thead>
<tr>
<th>Apple1</th>
<th>Apple2</th>
<th>Coconut1</th>
<th>Pear1</th>
<th>Strawberry1</th>
<th>Banana1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{N}(8, 0.5)$</td>
<td>$\mathcal{N}(7, 0.2)$</td>
<td>$\mathcal{N}(15, 0.75)$</td>
<td>$\mathcal{N}(13, 0.2)$</td>
<td>$\mathcal{N}(3.5, 1)$</td>
<td>$\mathcal{N}(17, 0.1)$</td>
</tr>
<tr>
<td>green-yellow</td>
<td>red</td>
<td>brown</td>
<td>green-yellow (3:1)</td>
<td>red</td>
<td>yellow</td>
</tr>
</tbody>
</table>

Table 2. Sample representatives used in the kitchen scenario, defined by a discrete color attribute and a continuous length attribute (in $cm$).

The considered example scenario evolves over 7 steps of discrete time. In the first time step $t = 1$, no representative is observed. In each of the time steps from
At time steps $t = 5$ and $t = 6$, one representative is observed, in the order as defined in Table 2, e.g., Apple1 at $t = 2$ and Pear1 at $t = 5$, with the exception of Banana1. In the so defined scenario, PMRs exist at time steps $t = 5$ and $t = 6$. At time step $t = 7$, a new concept Pear is added to $\mathcal{C}$, based on the attributes of Pear1. Fig. 5 shows the values of several measures resulting from this scenario, including the overall model quality $Q(\mathcal{R}, \mathcal{C})$, the model correspondence $L(\mathcal{R}|\mathcal{C})$, the association-based measure $M(p)$ (calculated just for the new representative in each time step) as well as the absolute and relative step measures $Q_{C,1}^a(\mathcal{R}, \mathcal{C})$ and $Q_{C,1}^r(\mathcal{R}, \mathcal{C})$. For calculating model correspondence and model quality, respectively, the Kullback Leibler divergence was used for rating attribute correspondence and (6) for rating attribute specificity.

As can be seen, all measures perform as expected. At the PMR time steps ($t = 5, 6$), description length increases, as reflected by model correspondence and quality. Simultaneously, the three detection measures identify Pear1 and Strawberry1 as PMRs. Adding the newly acquired concept Pear at $t = 7$ re-improves model correspondence and quality as expected.

![Graph](image)

**Fig. 5.** Evolution of several measures for model quality and PMR detection in the defined example kitchen scenario over 7 discrete steps of time.

### 6 Conclusion

In this contribution, an extension to adaptive knowledge management for the Object-Oriented World Model was presented. The extension proposes means for detecting and identifying points of necessary change in a probabilistic knowledge model, i.e., poorly mapped representatives. Different local and global measures
for detecting such representatives were presented and evaluated. In addition, an approach to model adaptation based on those representatives was proposed.

A further elaboration and an extensive evaluation of the proposed ideas and their interplay in a system for adaptive knowledge management is subject to future work. This comprises evaluating the approach based on a realistic scenario using e.g. data acquired by computer vision techniques.

References

Verzeichnis der zuletzt erschienenen Informatik-Berichte

[351] Fechner, B.:  
   **GPUs for Dependability**

[352] Güting, R. H., Behr, T., Xu, J.:  
   **Efficient k-Nearest Neighbor Search on Moving Object Trajectories**

   **CCA 2009 Sixth International Conference on Computability and Complexity in Analysis**

[354] Beierle, C., Kern-Isberner, G.:  
   **Relational Approaches to Knowledge Representation and Learning**

[355] Sakr, M.A., Güting, R.H.:  
   **Spatiotemporal Pattern Queries**

[356] Güting, R. H., Behr, T., Düntgen, C.:  
   **SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations**

[357] Düntgen, C., Behr, T., Güting, R.H.:  
   **Assessing Representations for Moving Object Histories**

[358] Sakr, M.A., Güting, R.H.:  
   **Group Spatiotemporal Pattern Queries**

[359] Hartrumpf, S., Helbig, H., vor der Brück, T., Eichhorn, C.:  
   **SemDupl: Semantic Based Duplicate Identification**

[360] Xu, J., Güting, R.H.:  
   **A Generic Data Model for Moving Objects**

[361] Beierle, C., Kern-Isberner, G.:  
   **Evolving Knowledge in Theory and Application: 3rd Workshop on Dynamics of Knowledge and Belief, DKB 2011**

[362] Xu, J., Güting, R.H.:  
   **GMOBench: A Benchmark for Generic Moving Objects**

[363] Finthammer, M.:  
   **A Generalized Iterative Scaling Algorithm for Maximum Entropy Reasoning in Relational Probabilistic Conditional Logic Under Aggregation Semantics**

[364] Güting, R.H., Behr, T., Düntgen, C.:  
   **Book Chapter: Trajectory Databases**

[365] Paul, A.; Rettinger, R.; Weihrauch, K.:  
   **CCA 2012 Ninth International Conference on Computability and Complexity in Analysis (extended abstracts)**

[366] Lu, J., Güting, R.H.:  
   **Simple and Efficient Coupling of a Hadoop With a Database Engine**

[367] Hoyrup, M., Ko, K., Rettinger, R., Zhong, N.:  
   **CCA 2013 Tenth International Conference on Computability and Complexity in Analysis**