Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Minimal Tolerance Pairs for System Z-Like Ranking Functions for First-Order Conditional Knowledge Bases

Christoph Beierle
University of Hagen
Hagen, Germany

Tobias Falke
University of Hagen
Hagen, Germany

Steven Kutsch
University of Hagen
Hagen, Germany

Gabriele Kern-Isberner
Univ. of Technology Dortmund
Dortmund, Germany

Abstract

In system Z, reasoning is done with respect to a unique minimal ranking function obtained from a partitioning of the conditionals in a knowledge base. In this paper, we extend system Z^{FO}, a recent proposal for a system Z-like approch to first-order conditionals. We introduce the notion of tolerance pair and show how sceptical Z^{FO}-inference can be defined and implemented by taking all minimal tolerance pairs into account.

1 Introduction

In nonmonotonic reasoning over a conditional knowledge base $\mathcal{R}$, system Z (Pearl 1990) is a well-established approach where entailment for $\mathcal{R}$ is defined with respect to an ordinal conditional (OCF) (Spohn 2012) obtained from a unique partitioning of the conditionals in $\mathcal{R}$. System Z has spawned a number of extensions (e.g. (Goldszmidt, Morris, and Pearl 1993; Bourne and Parsons 1999)). Recently, a proposal for a system Z^{FO} with system Z-like OCFs for first-order conditionals has been made (Kern-Isberner and Beierle 2015), based on the first-order conditional semantics using OCFs developed in (Kern-Isberner and Thimm 2012). In (Kern-Isberner and Beierle 2015), certain partitionings of the conditionals in $\mathcal{R}$ and of the constants are assumed to be given. This paper extends that approach in several directions. We introduce the notion of tolerance pair, define sceptical Z^{FO}-inference with respect to all OCFs obtained from minimal tolerance pairs, and develop an algorithm computing exactly all minimal tolerance pairs. We also give an overview of the software system ZIFO that implements system Z^{FO} and Z^{FO}-inference.

2 Background: OCFs and System Z

In propositional settings, *ordinal conditional functions* (OCF, (Spohn 2012)), also called *ranking functions* are a well-known framework for nonmonotonic reasoning and belief revision. We recall very briefly the basic details of this approach and System Z (Pearl 1990; Goldszmidt and Pearl 1996). Let $\mathcal{L}_\Sigma$ be a propositional language over a set Σ of propositional atoms. Let Ω denote the set of possible worlds over $\mathcal{L}$; Ω will be taken simply as the set of all propositional interpretations over $\mathcal{L}$ and can be identified with the set of all complete conjunctions over Σ. For $\omega \in \Omega$, $\omega \models A$ means that the propositional formula $A \in \mathcal{L}_\Sigma$ holds in the possible world ω. A (propositional) conditional $(B|A)$ is an object of a three-valued nature, partitioning the set of worlds Ω in three parts: those worlds satisfying AB, thus *verifying* the conditional, those worlds satisfying $A\overline{B}$, thus *falsifying* the conditional, and those worlds not fulfilling the premise A and so which the conditional may not be applied to at all.

An *ordinal conditional function* is a function $\kappa : \Omega \to \mathbb{N} \cup \{\infty\}$ with $\kappa^{-1}(0) \neq \emptyset$ which maps each world $\omega \in \Omega$ to a degree of implausibility $\kappa(\omega)$; ranks of formulas $A \in \mathcal{L}_\Sigma$ are defined by $\kappa(A) = \min\{\kappa(\omega) \mid \omega \models A\}$ with $\min(\emptyset) = \infty$. An OCF κ *accepts* a conditional $(B|A)$, in symbols $\kappa \models (B|A)$, if and only if $\kappa(AB) < \kappa(A\overline{B})$, that is, if and only if the conditional's verification AB is more plausible than its falsification $A\overline{B}$. In this case, we call κ a (ranking) model of $(B|A)$, and κ is a (ranking) model of a conditional knowledge base $\mathcal{R}$ if it is a model of each of the conditionals in $\mathcal{R}$.

For a given conditional knowledge base $\mathcal{R} = \{(B_1|A_1), \ldots, (B_n|A_n)\}$, the system Z approach by Pearl (Pearl 1990; Goldszmidt and Pearl 1996) defines an OCF κ_z that is a model of $\mathcal{R}$ and that is unique among all such models in that it restricts the plausibility of worlds in a minimal way. System Z is based on a notion of tolerance: A conditional $(B|A)$ 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 $1 \leqslant i \leqslant n$, i.e. iff ω verifies $(B|A)$ and does not falsify any $(B_i|A_i)$. Now, system Z is set up by first partitioning $\mathcal{R} = \mathcal{R}_0 \cup \ldots \cup \mathcal{R}_m$ into maximal sets $\mathcal{R}_j$ such that each conditional in $\mathcal{R}_j$ is tolerated by $\cup_{i \geqslant j} \mathcal{R}_i$. Then the function $Z : \mathcal{R} \to \mathbb{N}$ is defined by $Z(B|A) = k$ iff $(B|A) \in \mathcal{R}_k$, and finally, κ_z is given by

$$\kappa_z(\omega) = \begin{cases} 0, & \text{iff } \omega \models (A_i \Rightarrow B_i) \text{ for all } 1 \leqslant i \leqslant n \\ \max\limits_{1 \leqslant i \leqslant n} \{Z(B_i|A_i) \mid \omega \models A_i\overline{B_i}\} + 1, & \text{otherwise.} \end{cases}$$

3 OCFs for First-Order Conditionals

We recall the basics of system Z^{FO} (Kern-Isberner and Beierle 2015). Let Σ be a first-order signature consisting of a finite set of predicates P_Σ and a finite set of constant symbols D_Σ but without function symbols of arity > 0. An *atom* is a predicate of arity n together with a list of n constants and/or variables. A *literal* is an atom or a negated atom. Formulas are built on atoms using conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and quantification ($\forall$, $\exists$). We abbreviate conjunctions by juxtaposition and negations usually by overlining, e. g. AB means $A \wedge B$ and $\overline{A}$ means $\neg A$. A *ground* formula contains no variables. In a *closed* formula, all variables (if they occur) are bound by quantifiers, otherwise, the formula is *open*, and the variables that occur outside of the range of quantifiers are called *free*. If a formula A contains free variables we also use the notation $A(\vec{x})$ where $\vec{x} = (x_1, \ldots, x_n)$ contains all free variables in A. If $\vec{c}$ is a vector of the same length as $\vec{x}$ then $A(\vec{c})$ is meant to denote the instantiation of A with $\vec{c}$. A formula $\forall\vec{x}A(\vec{x})$ ($\exists\vec{x}A(\vec{x})$) is *universal* (*existential*) if A involves no further quantification. Let $\mathcal{L}_\Sigma$ be the first-order language that allows no nested quantification, i.e., all quantified formulas are either universal or existential formulas.

$\mathcal{L}_\Sigma$ is extended by a conditional operator "$|$" to a conditional language $(\mathcal{L}_\Sigma \,|\, \mathcal{L}_\Sigma)$ containing first-order conditionals $(B \,|\, A)$ with $A, B \in \mathcal{L}_\Sigma$, and (universally or existentially) quantified conditionals $\forall\vec{x}(B \,|\, A)$, $\exists\vec{x}(B \,|\, A)$[1]. When writing $(B(\vec{x}) \,|\, A(\vec{x}))$, we assume $\vec{x}$ to contain all free variables occurring in either A or B. For $r = (B \,|\, A) \in (\mathcal{L}_\Sigma \,|\, \mathcal{L}_\Sigma)$ (without outer quantification), we set $\overline{r} = (\overline{B} \,|\, A)$. Conditionals cannot be nested. When the signature is clear from context, we may omit the subscript Σ.

Definition 1. *A* first-order knowledge base $\mathcal{KB} = \langle \mathcal{F}, \mathcal{R} \rangle$ *(over Σ) consists of a finite set of conditionals $\mathcal{R}$ from $(\mathcal{L}_\Sigma \,|\, \mathcal{L}_\Sigma)$ with the restriction that no existential (outer) quantification of conditionals may occur, together with a set $\mathcal{F}$ of closed formulas from $\mathcal{L}_\Sigma$, called* facts*.*

In this way, we can accurately distinguish between the statements "A certainly holds for all individuals" ($\forall x A(\vec{x}) \in \mathcal{F}$), "it is plausible that A holds for all individuals" ($\forall x A(\vec{x}) \in \mathcal{R}$, treated as $(\forall x A(\vec{x}) \,|\, \top)$), and "$A$ is plausible" ($A(\vec{x}) \in \mathcal{R}$, treated as $(A(\vec{x}) \,|\, \top)$). In general, having a classical (i. e., unconditional) formula A in $\mathcal{F}$ expresses "A is certain" while A in $\mathcal{R}$ means "A is plausible".

Example 2 (Penguins and super-penguins)**.** *Assume that there are* penguins (P), birds (B)*, and* super-penguins (S) *as well as* winged things (W) *and* flying things (F). *Our universe $D = \{p, t, s\}$ consists of the objects resp. constants $t = Tweety, p = Polly, s = Supertweety$. The knowledge base $\mathcal{KB}_{pen} = \langle \mathcal{F}, \mathcal{R} \rangle$ consists of the facts $\mathcal{F} = \{B(p), P(t), S(s), \forall x S(x) \Rightarrow P(x), \forall x P(x) \Rightarrow B(x)\}$, and $\mathcal{R} = \{r_1, r_2, r_3, r_4\}$ containing four open first-order conditionals:*

$$r_1 : (F(x) \,|\, B(x)), \quad r_2 : (W(x) \,|\, B(x)),$$
$$r_3 : (\overline{F}(x) \,|\, P(x)), \quad r_4 : (F(x) \,|\, S(x)).$$

For classical interpretation of first-order aspects we use the Herbrand semantics. The *Herbrand base* $\mathcal{H}^\Sigma$ of a first-order signature Σ is the set of all ground atoms of Σ. A *possible world* ω is any subset of $\mathcal{H}^\Sigma$. Analogously to the propositional case, a possible world can be concisely represented as a *complete conjunction* or *minterm*, i. e. a conjunction of literals where every atom of $\mathcal{H}^\Sigma$ appears either in positive or in negated form. Also as in the propositional case, we denote the set of all possible worlds of Σ by Ω_Σ, and $\models$ denotes the classical satisfaction relation between possible worlds and first-order formulas from $\mathcal{L}_\Sigma$. Just as in the propositional case, the set Ω_Σ of possible worlds can be ranked by an ordinal conditional function $\kappa : \Omega_\Sigma \to \mathbb{N} \cup \{\infty\}$ with $\kappa^{-1}(0) \neq \emptyset$ that assigns degrees of implausibility resp. disbelief to possible worlds, and we will show how to extend κ to formulas in the first order case.

For an open conditional $r = (B(\vec{x}) \,|\, A(\vec{x})) \in (\mathcal{L}_\Sigma \,|\, \mathcal{L}_\Sigma)$, $\mathcal{H}^{(B(\vec{x}) \,|\, A(\vec{x}))}$ denotes the set of all constant vectors $\vec{a}$ used for proper groundings of $(B(\vec{x}) \,|\, A(\vec{x}))$ from $\mathcal{H}^\Sigma$, i. e. $\mathcal{H}^{(B(\vec{x}) \,|\, A(\vec{x}))} = D_\Sigma^{|\vec{x}|}$ where $|\vec{x}|$ is the length of $\vec{x}$. For $\vec{a} \in \mathcal{H}^{(B(\vec{x}) \,|\, A(\vec{x}))}$, let $r(\vec{a}) = (B(\vec{a}) \,|\, A(\vec{a}))$ denote the instantiation of r by $\vec{a}$. In the following, $A, B \in \mathcal{L}_\Sigma$ denote closed formulas, $A(\vec{x}), B(\vec{x}) \in \mathcal{L}_\Sigma$ denote open formulas.

Definition 3. *Let κ be an OCF. The κ-ranks of closed formulas are defined via*

$$\kappa(A) = \min_{\omega \models A} \kappa(\omega)$$

Furthermore, we define the κ-ranks for open formulas by evaluating most plausible instances:

$$\kappa(A(\vec{x})) = \min_{\vec{a} \in \mathcal{H}^{A(\vec{x})}} \kappa(A(\vec{a}))$$

The ranks of first-order formulas are coherently based on the usage of OCFs for propositional formulas. For the acceptance of conditionals, we first consider closed formulas.

Definition 4. *Let κ be an OCF. The acceptance relation $\models$ between κ and formulas from $\mathcal{L}_\Sigma$ and $(\mathcal{L}_\Sigma \,|\, \mathcal{L}_\Sigma)$ is*

- for closed formulas:
  - $\kappa \models A$ *iff for all $\omega \in \Omega$ with $\kappa(\omega) = 0$, it holds that $\omega \models A$.*
  - $\kappa \models (B \,|\, A)$ *iff* $\kappa(AB) < \kappa(A\overline{B})$.
- for universal/existential conditionals:
  - $\kappa \models \forall\vec{x}(B(\vec{x}) \,|\, A(\vec{x}))$ *iff* $\kappa \models (B(\vec{a}) \,|\, A(\vec{a}))$ *for all* $\vec{a} \in \mathcal{H}^{(B(\vec{x}) \,|\, A(\vec{x}))}$.
  - $\kappa \models \exists\vec{x}(B(\vec{x}) \,|\, A(\vec{x}))$ *iff there is* $\vec{a} \in \mathcal{H}^{(B(\vec{x}) \,|\, A(\vec{x}))}$ *such that* $\kappa \models (B(\vec{a}) \,|\, A(\vec{a}))$.

Acceptance of a sentence by a ranking function is the same as in the propositional case for ground sentences, and interprets the classical quantifiers in a straightforward way.

The treatment of acceptance of open formulas expressing default statements like in "usually, if A holds, then B also holds" is more intricate. The basic idea is that such (conditional) open statements hold if there are individuals providing most convincing instances of the respective conditional. These so-called *representatives* should, of course, allow for the acceptance of the instantiated conditional (as in Definition 4) while most plausibly verifying the conditional. Moreover, representatives are expected to be least exceptional with respect to falsifying the conditional.

[1]These quantifications will often be distinguished as *outer* quantifications in the paper.

Definition 5. *Let $r = (B(\vec{x}) \mid A(\vec{x})) \in (\mathcal{L}_\Sigma \mid \mathcal{L}_\Sigma)$ be a non-quantified conditional involving open formulas from $\mathcal{L}_\Sigma$. We say that $\vec{a} \in \mathcal{H}^{(B(\vec{x}) \mid A(\vec{x}))}$ is a* weak representative *of r iff it satisfies the following conditions:*

$$\kappa(A(\vec{a})B(\vec{a})) = \kappa(A(\vec{x})B(\vec{x})) \quad (1)$$

$$\kappa(A(\vec{a})B(\vec{a})) < \kappa(A(\vec{a})\overline{B}(\vec{a})) \quad (2)$$

The set of weak representatives of r *is denoted by $WRep(r)$. We say that $\vec{a} \in \mathcal{H}^{(B(\vec{x}) \mid A(\vec{x}))}$ is a* (strong) representative *of r iff it is a weak representative of r and*

$$\kappa(A(\vec{a})\overline{B}(\vec{a})) = \min_{\vec{b} \in WRep(r)} \kappa(A(\vec{b})\overline{B}(\vec{b})). \quad (3)$$

The set of all representatives of r *is denoted by $Rep(r)$.*

(Weak) Representatives of a conditional are characterized by being the most general and least exceptional; in (Kern-Isberner and Beierle 2015), this is illustrated using the penguin scenario. The definition of acceptance of open conditionals is based on the notion of representatives as follows.

Definition 6. *Let κ be an OCF and $r = (B(\vec{x}) \mid A(\vec{x}))$ an open (non-quantified) conditional. Then κ accepts r, denoted by $\kappa \models r$, iff $Rep(r) \neq \emptyset$, and one of the two following (exclusive) conditions is satisfied:*

(Acc-1) $$\kappa(A(\vec{x})B(\vec{x})) < \kappa(A(\vec{x})\overline{B}(\vec{x})) \quad (4)$$

(Acc-2) $\kappa(A(\vec{x})B(\vec{x})) = \kappa(A(\vec{x})\overline{B}(\vec{x}))$*, and for all $\vec{a}_1 \in Rep((B(\vec{x}) \mid A(\vec{x})))$ and for all $\vec{a}_2 \in Rep((\overline{B}(\vec{x}) \mid A(\vec{x})))$, it holds that*

$$\kappa(A(\vec{a}_1)\overline{B}(\vec{a}_1)) < \kappa(A(\vec{a}_2)B(\vec{a}_2)). \quad (5)$$

The acceptance of an open conditional is based on the existence of a suitable $\vec{a}$ satisfying (2), i. e., on the acceptance of the propositional conditional $(B(\vec{a}) \mid A(\vec{a}))$ (note that $Rep((B(\vec{x}) \mid A(\vec{x}))) \neq \emptyset$ iff $WRep((B(\vec{x}) \mid A(\vec{x}))) \neq \emptyset$). However, conditions (1) and (2) alone are too weak to justify the acceptance of $(B(\vec{x}) \mid A(\vec{x}))$ since it might well be the case that there are $\vec{a}$ and $\vec{b}$ fulfilling (1) and (2) for $(B(\vec{x}) \mid A(\vec{x}))$ and $(\overline{B}(\vec{x}) \mid A(\vec{x}))$, respectively. This means that κ might accept both $(B(\vec{x}) \mid A(\vec{x}))$ and $(\overline{B}(\vec{x}) \mid A(\vec{x}))$, which would be counterintuitive. Hence, we need to make acceptance unambiguous by giving preference to one of the two conditionals. This can be done either by postulating (4) or (5). Condition (4) looks like a natural prerequisite for the acceptance of $(B(\vec{x}) \mid A(\vec{x}))$. However, in the birds scenario with penguins and super-penguins, equalities like $\kappa(A(\vec{x})B(\vec{x})) = \kappa(A(\vec{x})\overline{B}(\vec{x}))$ quite naturally arise since penguins are as normal non-flying birds as doves are normal flying birds. In this case, (5) again uses the idea of least exceptionality for specifying proper representatives; it makes $(B(\vec{x}) \mid A(\vec{x}))$ acceptable, as opposed to $(\overline{B}(\vec{x}) \mid A(\vec{x}))$, if the representatives of the first conditional less exceptionally violate the respective conditional than the representatives of the latter conditional.

Definitions 5 and 6 can be used to define acceptance of open non-conditional formulas $A(\vec{x})$ by considering them as conditionals with tautological antecedents, i.e., as $(A(\vec{x}) \mid \top)$. However, it is crucial to remark here that $(A(\vec{x}) \mid \top)$ mandatorily demands for a default reading like "being A is plausible", as opposed to "A certainly holds". This distinction is made by distinguishing between certain knowledge $\mathcal{F}$ (all elements here are closed formulas of $\mathcal{L}_\Sigma$) and default (conditional) beliefs in $\mathcal{R}$ which may involve both closed and open formulas.

Definition 7. *Let $\mathcal{KB} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a first-order knowledge base, and let κ be an OCF.*

1. κ accepts $\mathcal{R}$*, denoted by $\kappa \models \mathcal{R}$, iff $\kappa \models \varphi$ for all $\varphi \in \mathcal{R}$.*
2. κ accepts $\mathcal{KB}$*, denoted by $\kappa \models \mathcal{KB}$, iff $\kappa(\omega) = \infty$ for all $\omega \not\models \mathcal{F}$, and $\kappa \models \mathcal{R}$.*

If $\kappa \models \mathcal{KB}$ then we also say that κ is a model *of $\mathcal{KB}$. If there is no κ with $\kappa \models \mathcal{KB}$ then $\mathcal{KB}$ is* inconsistent*.*

4 Tolerance Pairs and Inference

In (Kern-Isberner and Beierle 2015), a system Z-like ranking function for a first-order knowledge base is constructed from a partitioning of its conditionals and constants. Here, we will introduce the notions of *tolerance pairs* and their minimality and develop an algorithm for computig them. This yields a contructive way for generating system Z-like ranking functions for first-order conditionals and in particular the set of all ranking functions induced by minimal tolerance pairs.

Convention: As in (Kern-Isberner and Beierle 2015), we consider a restricted form of knowledge bases. For the rest of this paper, let $\mathcal{KB} = \langle \mathcal{F}, \mathcal{R} \rangle$ be a first-order knowledge base over a language $\mathcal{L}_\Sigma$ whose signature Σ consists of a set of constants D and only unary predicates. The conditionals in $\mathcal{R}$ can either involve open or closed formulas; we may omit the (outer) quantification of conditionals, as no existential conditional may occur, and all universal conditionals can be replaced by the set of all instantiations.

Definition 8 (Tolerance Pair)**.** *Let $\mathcal{R}_p = (\mathcal{R}_0, \dots, \mathcal{R}_m)$ and $D_p = (D_0, \dots, D_m)$ be ordered partionings of $\mathcal{R}$ and D, respectively. Then $(\mathcal{R}_p, D_p)$ is a* tolerance pair *for $\mathcal{KB}$ and D iff for all i, for all $r \in \mathcal{R}_i$, there is $a \in D_i$ and $\omega \in \Omega$ with $\omega \models \mathcal{F}$ such that ω verifies $r(a)$ and ω does not falsify $r'(a')$ for all $r' \in \cup_{j \geqslant i} \mathcal{R}_j$ and all $a' \in D_i$. By abuse of notation we also call $\pi = \langle (\mathcal{R}_0, D_0), \dots, (R_m, D_m) \rangle$ a tolerance pair; by π_i we denote the tuple $(\mathcal{R}_i, D_i)$ of π, and by $\pi_{\mathcal{R},i}$ and $\pi_{D,i}$ its components.*

Example 9. *Given the knowledge base $\mathcal{KB}_{pen}$ from Example 2, the following is a tolerance pair for $\mathcal{KB}_{pen}$ and D:*

$$\pi^{(1)} = \langle (\{r_1, r_2\}, \{p\}), (\{r_3\}, \{t\}), (\{r_4\}, \{s\}) \rangle \quad (6)$$

The notion of tolerance pair transfers the idea of tolerance system Z is based on to the first-order setting. In (Kern-Isberner and Beierle 2015) a theorem is given showing that a tolerance pair induces a ranking function that accepts $\mathcal{R}$.

Theorem 10 ((Kern-Isberner and Beierle 2015), κ_z^π)**.** *Assume that $\mathcal{F}$ is consistent and that no formula in $\mathcal{F}$ or $\mathcal{R}$ mentions more than one constant or variable, and let $\pi = \langle (\mathcal{R}_0, D_0), \dots (R_m, D_m) \rangle$ be a tolerance pair. For $\omega \in \Omega, \omega \models \mathcal{F}$, and for $0 \leqslant i \leqslant m$, define*

$$\lambda_i(\omega) = \begin{cases} 0, & \text{if } r(a) \text{ is not falsified in } \omega \ \forall a \in D_i, \forall r \in \mathcal{R} \\ \max\limits_{a \in D_i} \max\limits_{r \in \mathcal{R}} \{ j \mid r \in \mathcal{R}_j, \omega \text{ falsif. } r(a)\} + 1, & \text{otherw.} \end{cases}$$

Then the OCF κ_z^π defined by $\kappa_z^\pi(\omega) = \infty$ for $\omega \not\models \mathcal{F}$, and

$$\begin{aligned} \kappa_z^\pi(\omega) &= \sum_{i=0}^{m} (m+2)^i \lambda_i(\omega) - \kappa_0 \\ \kappa_0 &= \min_{\omega \in \Omega} \sum_{i=0}^{m} (m+2)^i \lambda_i(\omega) \end{aligned} \tag{7}$$

for $\omega \models \mathcal{F}$, is a model of $\mathcal{KB}$, called the OCF induced by π.

Example 11. *For $\pi^{(1)}$ from Example 9 and the world $\omega_0 = B(p)\overline{F}(p)\overline{P}(p)\ \overline{S}(p)\overline{W}(p)B(t)\overline{F}(t)P(t)\ \overline{S}(t)\overline{W}(t)$ $B(s)F(s)P(s)S(s)\overline{W}(s)$, Theorem 10 gives us:*

$$\lambda_1(\omega_0) = 1 \qquad \lambda_2(\omega_0) = 1 \qquad \lambda_3(\omega_0) = 3 \qquad \kappa_0 = 36$$

$$\kappa_z^{\pi^{(1)}}(\omega_0) = (1*1) + (4*1) + (16*3) - 36 = 17$$

However, in (Kern-Isberner and Beierle 2015), no algorithm for constructing the partitionings needed in Theorem 10 is given. For developing such an algorithm, we introduce partition pairs as a generalization of tolerance pairs.

Definition 12 (Partition Pair)**.** *Let $\mathcal{R}_p = (\mathcal{R}_0, \dots, \mathcal{R}_m)$ and $D_p = (D_0, \dots, D_m)$ be ordered partionings of subsets $\mathcal{R}' \subseteq \mathcal{R}$ and $D' \subseteq D$, respectively. Then $(\mathcal{R}_p, D_p)$ is a* partition pair *for $\mathcal{KB}$ and D. Again, we also call $\pi = \langle(\mathcal{R}_0, D_0), \dots (R_m, D_m)\rangle$ a partition pair.*

For $\mathcal{R}' = \{r_1, r_3\} \subseteq \mathcal{R}$ and $D' = \{p, t\} \subseteq D$ from Ex. 2, $\pi = \langle(\{r_1\}, \{p\}), (\{r_3\}, \{t\})\rangle$ is a partition pair.

Given a partition pair $\pi = \langle \pi_0, \dots, \pi_m \rangle$ for a given knowledge base, which partitions the subsets $\mathcal{R}' \subseteq \mathcal{R}$ and $D' \subseteq D$, there are several ways to extend it to new partition pairs. As the pairs will be constructed incrementally, starting with $\langle(\emptyset, \emptyset)\rangle$, we only add new elements to the highest tuple π_m in each step. The following extensions are possible:

1. *Add a conditional:* For each conditional $r_i \in (\mathcal{R} \setminus \mathcal{R}')$, with $1 \leqslant i \leqslant n, n = |\mathcal{R} \setminus \mathcal{R}'|$, a new partition pair $\pi' = \langle \pi_0, \dots, (\pi_{\mathcal{R},m} \cup \{r_i\}, \pi_{D,m})\rangle$ can be created.
2. *Add a constant:* For each constant $a_i \in (D \setminus D')$, with $1 \leqslant i \leqslant n, n = |D \setminus D'|$, a new partition pair $\pi' = \langle \pi_0, \dots, (\pi_{\mathcal{R},m}, \pi_{D,m} \cup \{a_i\})\rangle$ can be created.
3. *Enlarge the partition pair to $m+1$:* A new tuple $\pi_{m+1} = (\emptyset, \emptyset)$ can be added, yielding $\pi' = \langle \pi_0, \dots, \pi_m, (\emptyset, \emptyset)\rangle$. As we need to have every set $\pi_{\mathcal{R},i}$ and $\pi_{D,i}$ for $0 \leqslant i \leqslant m$ filled in a partition pair, this type of extension is only applicable if $\pi_{\mathcal{R},m}$ and $\pi_{D,m}$ both already contain at least one element. This makes sure that only the highest-labeled tuple, π_{m+1} after this extension, can temporarily have empty sets.
4. *Add a conditional and a constant:* For each $(r_i, a_j) \in (\mathcal{R} \setminus \mathcal{R}') \times (D \setminus D')$, with $n_{\mathcal{R}} = |\mathcal{R} \setminus \mathcal{R}'|$ and $n_D = |D \setminus D'|$, a new partition pair $\pi' = \langle \pi_0, \dots, (\pi_{\mathcal{R},m} \cup \{r_i\}, \pi_{D,m} \cup \{a_j\})\rangle$ can be created.

Algorithm 1, GENTP, generates every tolerance pair for a given knowledge base by incrementally extending partition pairs by performing one of these extension steps. Each time a partition pair has been expanded with a new component $(\emptyset, \emptyset)$ in step 3, step 4 is performed since it does not make sense to perform just step 1 or step 2. If the subalgorithm TEST$(\pi, i, \mathcal{R}_L, D_L)$ determines that the current partition pair can not be expanded to a tolerance pair, this path is pruned from the search space. In the following, we will present a refinement of TEST$_{\text{MIN}}$ ensuring that only minimal tolerance pairs are generated.

Algorithm 1 GENTP

Input: $\mathcal{KB} = \langle \mathcal{F}, \mathcal{R} \rangle$ and D
Output: Set of tolerance pairs Π

$\Pi \leftarrow \emptyset$
$V \leftarrow \emptyset$
SEARCH$(\langle(\emptyset, \emptyset)\rangle, 0, \mathcal{R}, D)$

function SEARCH$(\pi, i, \mathcal{R}_L, D_L)$
  if not TEST$(\pi, i, \mathcal{R}_L, D_L)$ **then**
    return ▷ Π contains calculated tolerance pairs
  if $\pi_{\mathcal{R},i} = \emptyset$ **and** $\pi_{D,i} = \emptyset$ **then** ▷ extension (4)
    for $(r, a) \in \mathcal{R}_L \times D_L$ **do**
      $\pi' \leftarrow \pi$ with $\pi'_{\mathcal{R},i} \leftarrow \pi_{\mathcal{R},i} \cup \{r\}$ and $\pi'_{D,i} \leftarrow \pi_{D,i} \cup \{a\}$
      SEARCH$(\pi', i, \mathcal{R}_L \setminus \{r\}, D_L \setminus \{a\})$
  else
    for $r \in \mathcal{R}_L$ **do** ▷ extension (1)
      $\pi' \leftarrow \pi$ with $\pi'_{\mathcal{R},i} \leftarrow \pi_{\mathcal{R},i} \cup \{r\}$
      SEARCH$(\pi', i, \mathcal{R}_L \setminus \{r\}, D_L)$
    for $a \in D_L$ **do** ▷ extension (2)
      $\pi' \leftarrow \pi$ with $\pi'_{D,i} \leftarrow \pi_{D,i} \cup \{a\}$
      SEARCH$(\pi', i, \mathcal{R}_L, D_L \setminus \{a\})$
  if $\pi_{\mathcal{R},i} \neq \emptyset$ and $\pi_{D,i} \neq \emptyset$ and $\mathcal{R}_L \neq \emptyset$ and $D_L \neq \emptyset$ **then** ▷ extension (3)
    $\pi' \leftarrow \langle \pi_0, \dots, \pi_i, (\emptyset, \emptyset)\rangle$
    SEARCH$(\pi', i+1, \mathcal{R}_L, \mathcal{D}_L)$

Definition 13 (Minimal Tolerance Pair, $\prec$)**.** *Let Π denote the set of possible tolerance pairs for $\mathcal{KB}$ and D. For $\pi_1 \in \Pi$ with m_1 partitions and $\pi_2 \in \Pi$ with m_2 partitions we say that π_1 is* smaller than *π_2, denoted by $\pi_1 \prec \pi_2$, iff:*

1. $m_1 < m_2$, or
2. $m_1 = m_2 = m$ and there is $0 \leqslant i \leqslant m$ with either $|(\pi_1)_{\mathcal{R},i}| > |(\pi_2)_{\mathcal{R},i}|$ or $|(\pi_1)_{\mathcal{R},i}| = |(\pi_2)_{\mathcal{R},i}|$ and $|(\pi_1)_{D,i}| > |(\pi_2)_{D,i}|$, such that for all $0 \leqslant j < i$, $|(\pi_1)_{\mathcal{R},j}| = |(\pi_2)_{\mathcal{R},j}|$ and $|(\pi_1)_{D,j}| = |(\pi_2)_{D,j}|$.

A tolerance pair π is minimal *for $\mathcal{KB}$ and D iff there is no $\pi' \in \Pi$ with $\pi' \prec \pi$.*

Tolerance pair $\pi^{(1)}$ (Ex. 9) is minimal with respect to $\prec$.

In system Z, an algorithm finds a partition of $\mathcal{R}$ in which every $r \in \mathcal{R}_i$ is tolerated by the set $\cup_{j=i}^{m} \mathcal{R}_j$ (Pearl 1990). In general, several partitions can exist that have this property. Among them, the algorithm finds a unique one in which every subset of the partition, from 0 to m, contains as many conditionals as possible. This minimal partition is then used to compute the ranking function κ_z. Extending this to the relational case of system Z$^{\text{FO}}$, a minimal tolerance pair ensures that from tuple π_0 to π_m, as many conditionals and as

Algorithm 2 $\textsc{Test}_{\textsc{min}}$

1: **function** $\textsc{Test}_{\textsc{min}}(\pi, i, \mathcal{R}_L, D_L)$
2:   **if** $\pi_{\mathcal{R},i} = \emptyset$ **and** $\pi_{D,i} = \emptyset$ **then** ▷ no tests for empty tuple
3:     **return** `true`
4:   **if** $\pi \in V$ **then** ▷ node already expanded?
5:     **return** `false`
6:   **else**
7:     $V \leftarrow V \cup \{\pi\}$
8:   **if** $\Pi \neq \emptyset$ **and** $\pi \succ_{PP} \pi_{min} \in \Pi$ **then** ▷ cannot be a minimum
9:     **return** `false`
10:   **if not** $\textsc{PotentialTP}(\pi, \mathcal{F}, \mathcal{R}_L)$ **then**
11:     **return** `false`
12:   **if** $\mathcal{R}_L = \emptyset$ **and** $D_L = \emptyset$ **then** ▷ tolerance pair found
13:     **if** $\Pi = \emptyset$ **or** $\pi \prec \pi_{min} \in \Pi$ **then** ▷ new minimum
14:       $\Pi \leftarrow \{\pi\}$
15:     **else if** $\pi \not\prec \pi_{min} \in \Pi, \pi_{min} \not\prec \pi$ **then** ▷ additional
16:       $\Pi \leftarrow \Pi \cup \{\pi\}$ ▷ minimum
17:     **return** `false`
18:   **return** `true`

many constants as possible are placed in the subsets.

System Z chooses the unique minimal partition since it induces the ranking with minimal degrees of implausibility. Thus, unless we have explicit information of the contrary, we assume every world to be as plausible as possible.

Since in contrast to system Z, there is not a unique minimal tolerance pair in system Z^{FO} for every knowledge base, we are interested in the set of all minimal tolerance pairs. Let the algorithm $\textsc{GenTP}_{\textsc{min}}$ be the algorithm obtained by replacing the call to $\textsc{Test}$ in line 5 of $\textsc{GenTP}$ by a call to the algorithm $\textsc{Test}_{\textsc{min}}$ (Algorithm 2). In line 10, the function $\textsc{PotentialTP}$ checks whether the current partition pair satisfies the condition in Definition 8 and is a potential tolerance pair. In lines 12 to 18 of $\textsc{Test}_{\textsc{min}}$, if a partition pair is found, it is checked whether it is smaller or equal to a pair π_{min} already found. If equal, it is added to the set of minimal partition pairs. If it is smaller than the pairs found so far, they can be discarded and only the new one is placed in Π. In addition, line 8 of $\textsc{Test}_{\textsc{min}}$ implements an additional pruning strategy. During the search, the pairs in set Π are the smallest partition pairs found so far. If the currently processed node represents a partition pair that is already bigger, its subtree can be safely skipped. The comparison $\succ_{PP}$ used here is similar to $\succ$ (given indirectly by Def. 13), with the only difference that the highest-labeled subsets are not taken into account since they are not yet final for π and could have additional elements added. Thus, a subtree is skipped if the current node's partition pair already has more tuples than the current minimum or if one subset $\pi_{\mathcal{R},i}$ or $\pi_{D,i}$ is smaller than the corresponding subset of the current minimum, with i ranging from 0 to $m-1$. This kind of pruning obviously does not influence the completeness of the search, yielding the observation that Algorithm $\textsc{GenTP}_{\textsc{min}}$ is sound and complete, i.e., that it generates exactly all minimal tolerance pairs for $\mathcal{KB}$ and D.

System Z yields a unique partitioning of the set of conditionals and thus a unique OCF that is used for inference in system Z. The next definition transfers this concept to system Z^{FO} by taking all minimal tolerance pairs into account.

Definition 14 (Z^{FO}-inference, $\mathrel{|\!\!\sim}_{\mathrm{z^{FO}}}$)**.** *Let $\mathcal{KB}$ ba a knowldeg base, A, B formulas and κ an OCF. We say that A κ-*entails *B (written $A \mathrel{|\!\!\sim}^{\kappa} B$) iff $\kappa \models (B|A)$. B is* a (sceptical) Z^{FO}-inference from A in the context of $\mathcal{KB}$*, denoted by $A \mathrel{|\!\!\sim}^{\mathcal{KB}}_{\mathrm{z^{FO}}} B$, iff for all minimal tolerance pairs π of $\mathcal{KB}$, we have $A \mathrel{|\!\!\sim}^{\kappa} B$ where $\kappa = \kappa^{\pi}_{\mathrm{z}}$.*

Section 5 illustrates Z^{FO}-inference with some examples.

5 Implementing System Z^{FO}

The software system ZIFO implements system Z^{FO}: For any $\mathcal{KB}$ and D, it can compute tolerance pairs and their induced ranking functions, and it can compute the corresponding inference relation induced by these ranking functions.

Figure 1 shows the user interface of system Z^{FO}. It consists of three main parts: a tool bar on the top and an input and an output area in the bottom left and right. The input area in Figure 1 shows the knowledge base $\mathcal{KB}_{pen}$ from Example 2 after automatic translation into ZIFO's internal syntax where universal quantification in $\mathcal{F}$ has been replaced by groundings and formulas are expressed using only negation (`!`), conjunction (`,`) and disjunction (`;`). Once a knowledge base has been loaded, tolerance pairs can be computed using three different algorithms (brute force, $\textsc{GenTP}$, $\textsc{GenTP}_{\textsc{min}}$). The output area in Figure 1 shows all three tolerance pairs that exist for $\mathcal{KB}_{pen}$, the only difference among them being that $(W(x)|B(X))$ is placed in another subset; only the first tolerance pair is a minimal one.

After selecting a tolerance pair π, system Z^{FO} offers option for computing the induced OCF $\kappa^{\pi}_{\mathrm{z}}$, given a detailed explanation for the derivation of $\kappa^{\pi}_{\mathrm{z}}$, or for storing $\kappa^{\pi}_{\mathrm{z}}(\omega)$ for all $\omega \in \Omega$ in a *CSV* file. Consider the following output:

```
Selected tolerance pair:
Pair 1 (m=2)
0 --- [(F(X)|B(X)), (W(X)|B(X))] --- [p]
1 --- [(!F(X)|P(X))] --- [t]
2 --- [(F(X)|S(X))] --- [s]
World:
w = (B(p)=1 B(t)=1 B(s)=1 P(p)=0 P(t)=1
     P(s)=1 S(p)=0 S(t)=0 S(s)=1 F(p)=0
     F(t)=0 F(s)=0 W(p)=0 W(t)=0 W(s)=0)
Satisfies facts: true
Ranking: 17
lambda(0,w) = 1
lambda(1,w) = 1
lambda(2,w) = 3
k_0: 36
k(w) = 1 * 1 + 4 * 1 + 16 * 3 - 36 = 17
```

We can see the tolerance pair and the selected world. The world satisfies the facts and is ranked with 17. The values λ_i for each $i \in \{0, \ldots, m\}$ as well as κ_0 are shown, leading to the computation shown in the last line.

By testing the acceptance of conditionals, ZIFO implements the reasoning behavior of system Z^{FO}-entailment. In standard mode, the program only returns `true` or `false` to

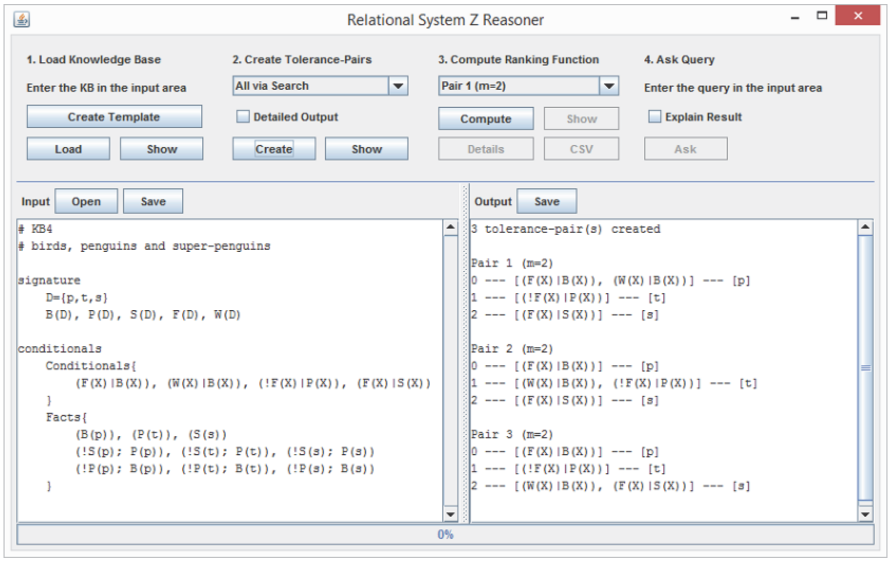

Figure 1: User interface of ZIFO implementing system Z^{FO}

indicate the acceptance of the query, and in *Explain Result* mode, we can make the program show us the whole evaluation process. For instance, consider the ranking function induced by the first tolerance pair shown in Figure 1. Using the conditional `(F(p)|B(p))` we can ask whether our bird Polly usually flies. In *Explain Result* mode, ZIFO returns:

```
0: Acceptance by ranking function k
  1: k |= (F(p)|B(p)) -> true
   2: k( B(p) F(p) ) = 0
    3: (B(p)=1 B(t)=1 B(s)=1 P(p)=0 P(t)=1
        P(s)=1 S(p)=0 S(t)=0 S(s)=1 F(p)=1
        F(t)=0 F(s)=1 W(p)=1 W(t)=0 W(s)=0)
   2: k( B(p) !F(p) ) = 1
    3: (B(p)=1 B(t)=1 B(s)=1 P(p)=0 P(t)=1
        P(s)=1 S(p)=0 S(t)=0 S(s)=1 F(p)=0
        F(t)=0 F(s)=1 W(p)=0 W(t)=0 W(s)=0)
   2: 0 < 1 ?
```

We can see that the conditional is accepted. First, for both the verifying formula $B(p)F(p)$ and the falsifying formula $B(p)\overline{F}(p)$, the ranks are derived by finding the most plausible world satisfying the formula. By comparing both ranks, the acceptance can be determined. The program provides the full evaluation tree down to the most basic definition.

Since the chosen tolerance pair is the only minimal one for $\mathcal{KB}_{pen}$, we thus have $B(p) \mathrel{\mid\!\sim}^{\mathcal{KB}_{pen}}_{\mathrm{Z}^{\mathrm{FO}}} F(p)$. To illustrate Z^{FO}-inference for the case of multiple minimal tolerance pairs, let $\mathcal{KB}'_{pen}$ be the knowledge base obtained from $\mathcal{KB}_{pen}$ by removing the two facts $B(p)$ and $P(t)$. $\mathcal{KB}'_{pen}$ has two different minimal tolerance pairs, but still $B(p) \mathrel{\mid\!\sim}^{\mathcal{KB}'_{pen}}_{\mathrm{Z}^{\mathrm{FO}}} F(p)$ holds. We also have $B(t) \mathrel{\mid\!\sim}^{\mathcal{KB}'_{pen}}_{\mathrm{Z}^{\mathrm{FO}}} F(t)$ which is a plausible inference since in $\mathcal{KB}'_{pen}$, Tweety is not known to be a penguin.

6 Conclusions and Further Work

For a system Z-like approach for first-order conditionals, we introduced the notion of tolerance pairs and presented a software system computing all minimal tolerance pairs for performing sceptical Z^{FO}-inference. Our current work includes investigating the complexity and formal properties of Z^{FO}-inference and developing a precise characterization of the consistency of a Z^{FO} knowledge base.

References

Bourne, R., and Parsons, S. 1999. Maximum entropy and variable strength defaults. In *Proceedings Sixteenth International Joint Conference on Artificial Intelligence, IJCAI'99*, 50–55.

Goldszmidt, M., and Pearl, J. 1996. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artif. Intell.* 84:57–112.

Goldszmidt, M.; Morris, P.; and Pearl, J. 1993. A maximum entropy approach to nonmonotonic reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(3):220–232.

Kern-Isberner, G., and Beierle, C. 2015. A system Z-like approach for first-order default reasoning. In *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation*, volume 9060 of *LNAI*. Springer. 81–95.

Kern-Isberner, G., and Thimm, M. 2012. A ranking semantics for first-order conditionals. In *ECAI-2012*, 456–461. IOS Press.

Pearl, J. 1990. System Z: A natural ordering of defaults with tractable applications to nonmonotonic reasoning. In *Proc. of the 3rd conf. on Theor. asp. of reasoning about knowledge*, TARK '90, 121–135. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Spohn, W. 2012. *The Laws of Belief: Ranking Theory and Its Philosophical Applications*. Oxford University Press.