Similarity Measures between Order-Sorted Logical Arguments

Victor David[1] **Jérôme Delobelle**[2] **Jean-Guy Mailly**[2]

[1] Department of Mathematics and Computer Science University of Perugia, Italy
[2] Université Paris Cité, LIPADE, F-75006 Paris, France
victor.david@unipg.it
jerome.delobelle@u-paris.fr
jean-guy.mailly@u-paris.fr

Abstract

Similarity in formal argumentation has received some attention recently, since one can argue that, in some context, using similar arguments to reach a conclusion is not the same as using dissimilar ones. In this work, we adapt the notion of similarity measures to arguments built from Order-Sorted First Order Logic, an extension of First Order Logic which allows to represent complex information, considering the type of the data. We study and evaluate our approach with respect to an adaptation of axioms from the literature. This paves the way to new reasoning modes taking into account similarity between arguments in complex settings like ontologies.

1 Introduction

Formal argumentation has become a major topic in Knowledge Representation and Reasoning (KRR), with various applications like decision making [28], defeasible reasoning [16], dealing with inconsistent knowledge bases [12], as well as in multi-agent systems [23]. So, when agents use logic-based information for reasoning, it is possible to build arguments from this information, where typically an argument is a pair made of a set of formulae (called support) and a single formula (called conclusion). The conclusion should be a logical consequence of the support. Examples of arguments are $A = \langle\{p \wedge q \wedge r\}, p \wedge q\rangle$, $B = \langle\{p \wedge q\}, p \wedge q\rangle$ and $C = \langle\{p, q\}, p \wedge q\rangle$. From the definition of arguments, one can identify attacks between them, and then use a semantics to evaluate the arguments. Finally, conclusions of the "strong" arguments are inferred from the base. In the literature, there exist several families of semantics (e.g. extension-based, ranking-based or gradual semantics) to determine which arguments are "strong". We refer the reader to [1] for a recent overview of the existing families of semantics in abstract argumentation and the differences between these approaches (e.g., definition, outcome, application). Among the existing gradual semantics, like h-Categorizer [12], some of them satisfy the Counting (or Strict Monotony) principle defined in [2]. This principle states that each attacker of an argument contributes to weakening the argument. For instance, if the argument $D = \langle\{\neg p \vee \neg q\}, \neg p \vee \neg q\rangle$ is attacked by A, B, C, then each of the three arguments will decrease the strength of D. However, the three attackers are somehow similar, thus D will lose more than necessary. Consequently, the authors in [4] have motivated the need for investigating the notion of similarity between pairs of such logical arguments. They introduced a set of principles that a reasonable similarity measure should satisfy, and provided several measures that satisfy them. In [3, 5, 6] several extensions of h-Categorizer that take into account similarities between arguments have been proposed. All these works consider propositional logic. In this paper, we suggest to adapt the principles behind similarity measures for logical arguments to a much more expressive framework, namely Order-Sorted First Order Logic ($\mathtt{OS-FOL}$), a formalism which generalizes (standard) First Order Logic ($\mathtt{FOL}$). Fragments of $\mathtt{OS-FOL}$ have been used for reasoning in various domains (*e.g.* [17] uses FOL for reasoning about policies, and [22] proposes an architecture for building cognitive agents able of deduction on facts and rules inferred directly from natural language). More generally, many KRR formalisms can be captured through $\mathtt{OS-FOL}$, like Description Logics [11]. While $\mathtt{FOL}$ has already interesting modelling capabilities, $\mathtt{OS-FOL}$ allows to naturally model situations where variables belong to a given domain, and there can be relations between the domains of the variables (*e.g.*, the domains made of all the penguins is a subset of the domain containing all the birds). So, by studying logical arguments built from $\mathtt{OS-FOL}$, we are able to apply our work to existing argumentation frameworks based on FOL [13, 10], but also other rich frameworks like Description Logics [11], which can be translated into (Order-Sorted) FOL. This paves to way to applications of argumentation (and similarity measures) to inconsistent knowledge expressed in these rich structured frameworks.

2 Background

2.1 Logic and Arguments

We assume that the reader is familiar with propositional logic. First Order Logic (FOL) is a rich framework for expressing knowledge about objects, including relations between them (using predicates). An example is "Tweety is a penguin, all penguins are birds and all birds have wings, so Tweety has wings" which can be expressed as $penguin(Tweety) \land (\forall x, penguin(x) \rightarrow bird(x)) \land (\forall x, bird(x) \rightarrow haveWings(x))$ for the premises, and $haveWings(Tweety)$ as the consequence. However, this framework does not allow to distinguish between various types of objects. This means that it would be possible to write a FOL formula like $hasRoots(Tweety)$, which does not not make sense since Tweety is a bird, not a plant. Since we want to apply our method to contexts where data can have a specific type, we use Order-Sorted FOL, a generalization of (standard) FOL where all the variables are associated with a *sort* (as well as the parameters of the predicates). [1] Then, when interpreting a formula, the domain of variables is constrained by its sort. An additional constraint can be added to these sorts, as a partial order over them, corresponding to inclusion relations between the domains associated to the sorts.

Definition 1 (Order-Sorted FOL) *Let* $\mathbf{So} = \{s_1, \ldots, s_n\}$ *be a set of sorts, and* $\prec \subseteq \mathbf{So} \times \mathbf{So}$ *a partial order over* $\mathbf{So}$. *An* Order-Sorted First Order Language $\mathtt{OS-FOL}$*, is a set of formulae built up by induction from :*
– a set $\mathbf{C}$ *of constants (*$\mathbf{C} = \{a_1, \ldots, a_l\}$*),*
– a set $\mathbf{V}$ *of variables (*$\mathbf{V} = \{x^s, y^s, z^s, \ldots \mid s \in \mathbf{So}\}$*),*
– a set $\mathbf{P}$ *of predicates (*$\mathbf{P} = \{P_1, \ldots, P_m\}$*),*
– a function $\mathtt{ar} : \mathbf{P} \longrightarrow \mathbb{N}$ *which gives the arity of predicates,*
– a function $\mathtt{sort}$ *s.t. for* $P \in \mathbf{P}$, $\mathtt{sort}(P) \in \mathbf{So}^{\mathtt{ar}(P)}$*, and for* $c \in \mathbf{C}$, $\mathtt{sort}(c) \in \mathbf{So}$,
– the usual connectives ($\neg, \lor, \land, \rightarrow, \leftrightarrow$*), Boolean constants* $\top$ *(true) and* $\bot$ *(false) and quantifier symbols (*$\forall, \exists$*).*
A grounded formula *is a formula without any variable.*

We use lowercase greek letters (*e.g.* ϕ, ψ) to denote formulae, and uppercase ones (*e.g.* Φ, Ψ) to denote sets of formulae. The set of all formulae is denoted by $\mathtt{OS-FOL}$. We assume formulae to be *prenex*, *i.e.* written as $Q_1 x_1, \ldots, Q_k x_k \phi$ where Q_i is a quantifier (for each $i \in \{1, \ldots, k\}$) and ϕ is a non-quantified formula. A formula ϕ is in negative normal form (NNF) if and only if it does not contain implication or equivalence symbols, and every negation symbol occurs directly in front of an atom. Following [21], we slightly abuse words and denote by $\mathtt{NNF}(\phi)$ the formula in NNF obtained from ϕ by "pushing down" every occurrence of $\neg$ (using De Morgan's law) and eliminating double negations. For instance, $\mathtt{NNF}(\neg((P(a) \rightarrow Q(a)) \lor \neg Q(b))) = P(a) \land \neg Q(a) \land Q(b)$. In that case, we call *literal* either an atom (*i.e.* a predicate with its parameters) or the negation of an atom. We denote by $\mathtt{Lit}(\phi)$ the set of literals occurring in $\mathtt{NNF}(\phi)$, hence $\mathtt{Lit}(\neg((P(a) \rightarrow Q(a)) \lor \neg Q(b))) = \{P(a), \neg Q(a), Q(b)\}$. For a given set of predicates $\mathbf{P}$, we define $\mathbf{L} = \{P(x_1^{s_1}, \ldots, x_k^{s_k}), \neg P(x_1^{s_1}, \ldots, x_k^{s_k}) \mid P \in \mathbf{P}, \mathtt{sort}(P) = (s_1, \ldots, s_k)\}$ the set of literals. We say that a literal is *negative* when it starts with a negation, denoted by $\mathtt{Pol}(L) = -$. Otherwise we say that it is *positive*, denoted by $\mathtt{Pol}(L) = +$. And we say that two literals have the same *polarity* if they are either both positive or both negative.

Let $\phi \in \mathtt{OS-FOL}$, ϕ is in a conjunctive normal form (CNF) if it is a conjunction of clauses $\bigwedge_i cl_i$ where each clause cl_i is a disjunction of literals $\bigvee_j l_j$. For instance $P(a) \land (Q(a) \lor Q(b))$ is in a CNF while $(P(a) \land Q(a)) \lor Q(b)$ is not. CNF formulae are particular NNF formulae. Clauses are also usually represented as sets of literals.

In $\mathtt{OS-FOL}$, the partial order $\prec$ represents "sub-type" relations between groups of entities. For instance, the fact that dogs are a special type of mammals can be represented by such a sub-type relation. In the case where $s_1 \prec s_2$, a predicate which expects a parameter of type s_2 can be applied to a constant or variable of type s_1 (for instance, a predicate about mammals can be applied to dogs).

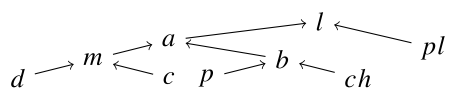

FIGURE 1 – Hierarchy of sorts from Example 1. An arrow from s_1 to s_2 means $s_1 \prec s_2$.

Example 1 $\mathtt{OS-FOL}$ *formulae can be used to reason about ontological information. Assume that we have the following information : mammals and birds are animals, dogs and cats are mammals, penguins and chickens are birds. Moreover, Zazu is a bird, Tweety is a penguin, and Dogmatix is a dog. Finally, animals are living beings, as well as plants. This can be represented by the following sorts and constants :*
– $\mathbf{So} = \{m, b, a, d, c, p, ch, l, pl\}$ *with* $m \prec a$, $b \prec a$, $d \prec m$, $c \prec m$, $p \prec b$, $ch \prec b$, $a \prec l$, $pl \prec l$ *(see Figure 1),*
– $Z \in \mathbf{C}$ *with* $\mathtt{sort}(Z) = b$ *is a constant for Zazu,*
– $T \in \mathbf{C}$ *with* $\mathtt{sort}(T) = p$ *is a constant for Tweety,*
– $D \in \mathbf{C}$ *with* $\mathtt{sort}(D) = d$ *is a constant for Dogmatix.*

We know that all birds have wings, and both mammals and birds are warm-blooded. Also, some birds and some

1. In this paper, we restrict ourselves to formulae without functions.

mammals fly, but not all of them. If a bird is wounded, then it cannot fly. If a bird is penguin, then it cannot fly. Some birds are wounded. Finally, Tweety is a penguin. This information can be represented by the predicates $\mathbf{P} = \{hW, wB, f, w, p\}$, *standing respectively for "haveWings", "warmBlooded", "fly", "wounded" and "penguin" s.t.* $\mathtt{ar}(P_i) = 1$ *and* $\mathtt{sort}(P_i) = a$ *for each* $P_i \in \mathbf{P}$.

We can build, e.g. *the formula* $\forall x^b, hW(x^b)$ *meaning that all birds have wings (because the variable* x^b *has the sort b). The other pieces of information are represented by*

$\forall x^b wB(x^b)$	$\forall x^m wB(x^m)$
$\exists x^b_1, x^b_2 f(x^b_1) \wedge \neg f(x^b_2)$	$\exists x^m_1, x^m_2 f(x^m_1) \wedge \neg f(x^m_2)$
$\forall x^b w(x^b) \rightarrow \neg f(x^b)$	$\forall x^b p(x^b) \rightarrow \neg f(x^b)$
$\exists x^b w(x^b)$	$p(T)$

However formulae like $\exists x^l, f(x^l)$ *or* $\forall x^{pl}, wB(x^{pl})$ *are not well-formed, since the predicates* fly *and* wB *cannot be applied to living beings or plants.*

OS − FOL formulae are evaluated via a notion of structure :

Definition 2 (Structure) *Given* $n \in \mathbb{N}$, *a* n-sorted structure *is* $\mathbf{St} = (Dom, Rel, Cons)$ *where :*
– $Dom = \{D_1, \ldots, D_n\}$ *are the (non-empty) domains,*
– $Rel = \{R_1, \ldots, R_m\}$ *are relations over the domains,*
– $Cons = \{c_1, \ldots, c_l\}$ *are constants in the domains.*

Example 2 *A structure associated with the* OS − FOL *from Example 1 is* $\mathbf{St} = (Dom, Rel, Cons)$ *where*
– $Dom = \{D_1 \ldots D_9\}$ *are the sets of all individuals of the various types (*e.g. D_1 *is the set of mammals, corresponding to the sort symbol* m *;* D_2 *is the set of birds, corresponding to the sort symbol* b *; etc),*
– $Rel = \{R_1, \ldots, R_5\}$ *are the relations corresponding to the predicate symbols (*e.g. R_1 *identifies winged animals,. . .)*
– $Cons = \{Zazu, Tweety, Dogmatix\}$ *are respectively a particular bird (an element of the domain* D_2 *associated with the sort* b*), a particular penguin (an element of the domain* D_6 *associated with the sort* p*) and a particular dog (an element of the domain* D_4 *associated with the sort* d*).*

Classical first order logic formulae can be evaluated via 1-sorted structures. For this reason, any fragment of first order logic is captured by OS − FOL. Now, we show how OS − FOL formulae are interpreted.

Definition 3 (Interpretation) *An* interpretation $\mathbf{I_{St}}$ over a structure $\mathbf{St}$ *assigns to elements of the* OS − FOL *vocabulary some values in the structure* $\mathbf{St}$. *Formally,*
– $\mathbf{I_{St}}(s_i) = D_i$, *for* $i \in \{1, \ldots, n\}$ *s.t. for each* $s_i, s_j \in \mathbf{So}$, *if* $s_i \preceq s_j$ *then* $\mathbf{I_{St}}(s_i) \subseteq \mathbf{I_{St}}(s_j)$ *(each sort symbol is assigned to a domain s.t. the sub-type relations are respected),*
– $\mathbf{I_{St}}(P_i) = R_i$, *for* $i \in \{1, \ldots, m\}$ *(each predicate symbol is assigned to a relation),*
– $\mathbf{I_{St}}(a_i) = c_i$, *for* $i \in \{1, \ldots, l\}$ *(each constant symbol is assigned to a constant value). As a shorthand, we write* $\mathbf{I_{St}}((s_1, \ldots, s_k)) = \mathbf{I_{St}}(s_1) \times \cdots \times \mathbf{I_{St}}(s_k)$. *Then satisfaction of formulae is recursively defined by :*
– $\mathbf{I_{St}} \models P_i(x_1, \ldots, x_k)$, *where* $(x_1, \ldots, x_k) \in \mathbf{I_{St}}((s_1, \ldots, s_k))$ *with* $\mathtt{sort}(x_i) = s_i$ *for each* $i \in \{1, \ldots, k\}$, *iff* $(x_1, \ldots, x_k) \in R_i$,
– $\mathbf{I_{St}} \models \exists x^{s_i} \phi$ *iff* $\mathbf{I}_{\mathbf{St}, x^{s_i} \leftarrow v} \models \phi$ *for some* $v \in D_i$,
– $\mathbf{I_{St}} \models \forall x^{s_i} \phi$ *iff* $\mathbf{I}_{\mathbf{St}, x^{s_i} \leftarrow v} \models \phi$ *for each* $v \in D_i$,
– $\mathbf{I_{St}} \models \phi \wedge \psi$ *iff* $\mathbf{I_{St}} \models \phi$ *and* $\mathbf{I_{St}} \models \psi$,
– $\mathbf{I_{St}} \models \phi \vee \psi$ *iff* $\mathbf{I_{St}} \models \phi$ *or* $\mathbf{I_{St}} \models \psi$,
– $\mathbf{I_{St}} \models \neg\phi$ *iff* $\mathbf{I_{St}} \not\models \phi$,
where $\mathbf{I}_{\mathbf{St}, x^{s_i} \leftarrow v}$ *is a modified version of* $\mathbf{I_{St}}$ *s.t. the variable* x^{s_i} *is replaced by a value* v *in the domain* D_i *corresponding to the sort symbol* s_i. *Finally, if* Φ *is a set of formulae, then* $\mathbf{I_{St}} \models \Phi$ *iff* $\mathbf{I_{St}} \models \phi$ *for each* $\phi \in \Phi$.

Observe that Definition 3 does not specify the satisfaction of implications and equivalences, but they can be defined as usual by $(\phi \rightarrow \psi) \equiv (\neg\phi \vee \psi)$, and $(\phi \leftrightarrow \psi) \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. We use $\mathtt{Mod}(\Phi)$ to denote the set of interpretations satisfying a set of formulae Φ, and we call Φ *consistent* if $\mathtt{Mod}(\Phi) \neq \emptyset$.

Example 3 *Continuing Example 1, we define* $\mathbf{I_{St}}$ *by :*
– $\mathbf{I_{St}}(m) = D_1$, $\mathbf{I_{St}}(b) = D_2, \ldots, \mathbf{I_{St}}(pl) = D_9$,
– $\mathbf{I_{St}}(hW) = R_1, \ldots, \mathbf{I_{St}}(p) = R_5$,
– $\mathbf{I_{St}}(Z) = Zazu$, $\mathbf{I_{St}}(T) = Tweety$, $\mathbf{I_{St}}(D) = Dogmatix$.
The formula $\phi = \forall x^b hW(x^b)$ *is satisfied by* $\mathbf{I_{St}}$, *since all elements of the domain* D_2 *associated with the sort symbol* b *actually have wings. On the contrary, consider the set of formulae* $\Phi = \{\forall x^b f(x^b), \forall x^p \neg f(x^p)\}$. *This set of formulae is not satisfied, because* $p \prec b$, *and so the domains satisfy* $D_6 \subset D_2$, *meaning that all penguins are birds. Then, from* Φ *we can deduce that any penguin can fly (because of the first formula) and cannot fly (because of the second formula) at the same time. So, this formula is not satisfied by* $\mathbf{I_{St}}$. *Notice that we could not define an interpretation* $\mathbf{I}'_{\mathbf{St}}$ *s.t.* $\mathbf{I}'_{\mathbf{St}}(Z) = Tweety$ *and* $\mathbf{I}'_{\mathbf{St}}(T) = Zazu$, *since Zazu is a bird, and* T *has the sort* p *(*i.e. *it can only be a penguin, not any kind of bird).*

Now we introduce the concept of instantiation, *i.e.* grounded formulae which are compatible with a given OS − FOL formula.

Definition 4 (Instantiation) *Given* Φ *a set of* OS − FOL *formulae and* $\mathbf{I_{St}}$ *an interpretation over a structure* $\mathbf{St}$, *the set of* instantiations *of* Φ *is defined recursively by :*
– $\mathtt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{\Phi\}$ *if* $\Phi = \{\phi\}$, *where* ϕ *is a grounded formula s.t.* $\mathbf{I_{St}} \models \phi$,
– $\mathtt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{\{\phi_{x^s \leftarrow v} \mid \mathbf{I_{St}} \models \phi_{x^s \leftarrow v}, v \in \mathbf{I_{St}}(s)\}\}$ *if* $\Phi = \{\forall x^s \phi\}$,
– $\mathtt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{\{\phi_{x^s \leftarrow v} \mid \mathbf{I_{St}} \models \phi_{x^s \leftarrow v}, v \in V\} \mid \emptyset \subset V \subseteq \mathbf{I_{St}}(s)\}$ *if* $\Phi = \{\exists x^s \phi\}$,
– $\mathtt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{I_1 \cup I_2 \mid I_1 \in \mathtt{Inst}_{\mathbf{I_{St}}}(\{\phi_1\}), I_2 \in$

$\texttt{Inst}_{\mathbf{I_{St}}}(\Phi_2), \mathbf{I_{St}} \models I_1 \cup I_2\}$ *if* $\Phi = \{\phi_1\} \cup \Phi_2$, *where* $\phi_{x^s \leftarrow v}$ *is the formula* ϕ *s.t. all the occurrences of the variable* x^s *are replaced by the value* v *(from the domain associated with the sort* s*).*

The idea is that formulae with quantified variables may be instanciated in various ways. Assuming that the domain of a variable x is $\{A, B\}$, then the formula $\exists x P(x)$ means that either $P(A)$ is true, or $P(B)$, or both at the same time. And $\forall x P(x)$ means that $P(A)$ and $P(B)$ are both true. This is what is captured by the notion of instantiation. Moreover, an instantiation is consistent because of the constraint $\mathbf{I_{St}} \models I_1 \cup I_2$ in the last part of the definition. This constraint means that, if *e.g.* we consider the set of formulae $\{\exists x P(x), \exists x \neg P(x)\}$, then we keep the instantiations where $P(A)$ is true and $P(B)$ is false, or the opposite. But we exclude situations where $P(A)$ is both true (because of the first formula) and false (because of the second formula) at the same time.

Example 4 *Consider the set of formulae* $\Phi = \{\phi_1 = \exists x^b w(x^b), \phi_2 = \forall x^b w(x^b) \rightarrow \neg f(x^b)\}$. *We assume here that the domain associated with the sort* b *is the set* $\{Tweety, Zazu\}$. *Applying Definition 4,* $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) = \{I_1 \cup I_2 \mid I_1 \in \texttt{Inst}_{\mathbf{I_{St}}}(\{\exists x^b w(x^b)\}), I_2 \in \texttt{Inst}_{\mathbf{I_{St}}}(\{\forall x^b w(x^b) \rightarrow \neg f(x^b)\}), \mathbf{I_{St}} \models I_1 \cup I_2\}$.

We start with the first formula, i.e. $\phi_1 = \exists x^b w(x^b)$. $\texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_1\}) = \{\{w(Tweety)\}, \{w(Zazu)\}, \{w(Tweety), w(Zazu)\}\}$. *For* $\phi_2 = \forall x^b w(x^b) \rightarrow \neg f(x^b)$, *there is only one possible instantiation :* $\texttt{Inst}_{\mathbf{I_{St}}}(\{\phi_2\}) = \{\{w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}\}$.

We conclude that $\texttt{Inst}_{\mathbf{I_{St}}}(\Phi) =$

$$\{\{w(Tweety), w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}, \{w(Zazu), w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}, \{w(Tweety), w(Zazu), w(Tweety) \rightarrow \neg f(Tweety), w(Zazu) \rightarrow \neg f(Zazu)\}\}$$

From the notions of structure and interpretation, we can define the consequence relation over $\texttt{OS} - \texttt{FOL}$ formulae.

Definition 5 (Consequence Relation) *Let* ϕ *and* ψ *be two* $\texttt{OS} - \texttt{FOL}$ *formulae. We say that* ψ is a consequence of ϕ, *denoted by* $\phi \vdash \psi$, *if for any structure* **St**, *and any interpretation* $\mathbf{I_{St}}$ *over* **St**, $\mathbf{I_{St}} \models \phi$ *implies* $\mathbf{I_{St}} \models \psi$. *Two formulae* ϕ, ψ *are* equivalent *(denoted* $\phi \equiv \psi$*) iff* $\phi \vdash \psi$ *and* $\psi \vdash \phi$.

Classical logic can be used to define arguments, *i.e.* logic-based representation of reasons supporting a specific conclusion. Logical arguments usually need to satisfy some constraints [12] :

Definition 6 (Logical Argument) *An* argument *built under a logic* $(\mathcal{L}, \vdash)$ *is a pair* $\langle \Phi, \phi \rangle$, *where* $\Phi \subseteq_f \mathcal{L}$ [2] *and* $\phi \in \mathcal{L}$, *s.t.* Φ *is consistent,* $\Phi \vdash \phi$, *and* $\nexists \Phi' \subset \Phi$ *s.t.* $\Phi' \vdash \phi$. *An argument* $A = \langle \Phi, \phi \rangle$ *is* trivial *iff* $\Phi = \emptyset$ *and* $\phi \equiv \top$. Φ *is called the support of the argument (*$\texttt{Supp}(A) = \Phi$*) and* ϕ *its conclusion (*$\texttt{Conc}(A) = \phi$*). The set of all arguments built under* $(\mathcal{L}, \vdash)$ *is denoted* $\texttt{Arg}(\mathcal{L})$.

2. $X \subseteq_f Y$ means X is a finite subset of Y

In this paper, we will focus on the set of arguments $\texttt{Arg}(\texttt{OS} - \texttt{FOL})$ built under the logic $(\texttt{OS} - \texttt{FOL}, \vdash)$, where $\vdash$ is the consequence relation from Definition 5.

Example 5 *Let* A_1 *and* A_2 *are examples of arguments :*

$$A_1 = \langle \{\exists x^b w(x^b), \forall x^b w(x^b) \rightarrow \neg f(x^b)\}, \exists x^b \neg f(x^b) \rangle$$
$$A_2 = \langle \{p(Tweety), \forall x^b p(x^b) \rightarrow \neg f(x^b)\}, \neg f(Tweety) \rangle$$

Note that two sets of formulae $\Phi, \Psi \subseteq_f \mathcal{L}$ are *equivalent*, denoted by $\Phi \cong \Psi$, iff there is a bijection $f : \Phi \rightarrow \Psi$ s.t. $\forall \phi \in \Phi$, $\phi \equiv f(\phi)$. However, we may want to consider that a set of formulae is equivalent with the conjunction of its elements (*e.g.* $\{P(a), Q(a)\}$ and $\{P(a) \wedge Q(a)\}$ are equivalent). For getting them equivalent, we borrow the method used in [7]. We transform every formula into a CNF, then we split it into a set containing its clauses. In our approach, we consider one CNF per formula. For that purpose, we will use a finite sub-language $\mathcal{F}$ that contains one formula per equivalent class and the formula should be in a CNF.

Definition 7 (Finite CNF Language $\mathcal{F}$) *Let* $\mathcal{F} \subset_f \mathcal{L}$ *s.t.* $\forall \phi \in \mathcal{L}$, *there is a unique* $\psi \in \mathcal{F}$ *s.t.* $\phi \equiv \psi$, $\texttt{Lit}(\phi) = \texttt{Lit}(\psi)$ *and* ψ *is a CNF formula. We define* $\texttt{CNF}(\phi) = \psi$.

While we do not specify the elements of $\mathcal{F}$, we use concrete formulae in the examples, and they are assumed to belong to $\mathcal{F}$.

Now we introduce $\texttt{UC}(\Phi)$ as the representation of the formulae in Φ as one set of clauses. Intuitively, recall that any formula can be seen as a set of clauses, associated with a sequence of quantifiers. A set of formulae can then be seen as set of clauses and a sequence of quantifiers, such that variables are renamed to avoid ambiguities. As an example, assume $\phi_1 = \exists x P(x) \wedge Q(x)$ and $\phi_2 = \exists x Q(x) \vee R(x)$. We have $\texttt{UC}(\{\phi_1, \phi_2\}) = \exists x, x' \{P(x), Q(x), Q(x') \vee R(x')\}$. Formally, for $\Phi = \{Q_{\phi_i} \phi_i \mid i \in \mathbb{N}\} \subseteq_f \mathcal{L}$, where ϕ_i is a non-quantified CNF formula (i.e. a set of clauses), and Q_{ϕ_i} is the sequence of quantifiers associated with ϕ_i, we define $\texttt{UC}(\Phi) = Q^*_{\phi_1} \dots Q^*_{\phi_n} \bigcup_{\phi \in \Phi} \bigcup_{\delta \in \phi} \delta^*$, where a renaming is applied to each clause (δ^*) and each sequence of quantifiers ($Q^*_{\phi_i}$) in order to guarantee that no variable is shared between quantifiers $Q^*_{\phi_i}$ and $Q^*_{\phi_j}$ (with $i \neq j$) or between clauses coming from different formulae ϕ_i and ϕ_j (with $i \neq j$). We simply write $\texttt{UC}(\phi)$ instead of $\texttt{UC}(\{\phi\})$, for $\phi \in \mathcal{L}$.

Note that $\texttt{UC}(\{P(a), Q(a)\}) \cong \texttt{UC}(P(a) \wedge Q(a))$.
Let us now introduce the notion of compiled argument.

Definition 8 (Compiled Argument) *The* compilation *of* $A \in \texttt{Arg}(\texttt{OS}-\texttt{FOL})$ *is* $A^* = \langle \texttt{UC}(\texttt{Supp}(A)), \texttt{Conc}(A)\rangle$.

We can see in the previous example that argument A is not concise, meaning that it has irrelevant information ($Q(b)$) for implying its conclusion. As it was shown in [7], using clausal arguments ensure that the arguments are concise.

Definition 9 (Equivalent Arguments) *Two arguments* $A, B \in \texttt{Arg}(\texttt{OS}-\texttt{FOL})$ *are equivalent, denoted by* $A \approx B$, *iff* $\texttt{UC}(\texttt{Supp}(A)) = \texttt{UC}(\texttt{Supp}(B))$ *and* $\texttt{UC}(\texttt{Conc}(A)) = \texttt{UC}(\texttt{Conc}(B))$. *We denote by* $A \not\approx B$ *when* A *and* B *are not equivalent.*

Definition 10 (Sub-argument) *Given two arguments* $A = \langle \Phi, \phi\rangle$ *and* $B = \langle \Psi, \psi\rangle$, *we say that* A *is a sub-argument of* B *if* $\texttt{UC}(\Phi) \subseteq \texttt{UC}(\Psi)$.

2.2 Binary Similarity Measure between OS − FOL Arguments

A similarity measure is used to indicate whether two arguments are similar or not, *i.e.* whether they share some parts of the reasoning mechanism used to build the arguments.

Definition 11 (Similarity Measure) *Let* $\mathbb{X}$ *be a set of objects. A similarity measure on* $\mathbb{X}$, *denoted by* $\texttt{sim}^{\mathbb{X}}$, *is a function from* $\mathbb{X} \times \mathbb{X}$ *to* $[0, 1]$.

In this section, we focus on similarity measures over arguments, *i.e.* $\mathbb{X} = \texttt{Arg}(\texttt{OS}-\texttt{FOL})$. Intuitively, $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, B)$ is close to 0 if the difference between A and B is important, while it is close to 1 if the arguments are similar. Several principles that similarity measures should satisfy have been discussed in the literature [4, 8, 7]. Some of the principles (Maximality, Symmetry, Substitution, and Syntax Independence) can be stated exactly as in the literature [7], since they do not concern the internal structure of the arguments. Notice that some authors have argued against the fact that a similarity measures should absolutely satisfy symmetry [27, 19]. For the other ones, we may need to adapt them to our OS − FOL-based arguments.

First, we adapt the Minimality principle. It states that, if two arguments do not have anything in common in their content, then their degree of similarity should be minimal. While, in propositional logic, determining the set of common propositional variables is enough, here we need to consider (domains of) predicates and constants. We do not consider variables here since they are use in the context of quantifiers : there is no reason to assume that there is something common between $\forall x, P(x)$ and $\forall x, Q(x)$.

Before presenting the Minimality principle, let us introduce some useful notations. Given a formula ϕ, $\texttt{Dom}(\phi) = \bigcup_{P \in \texttt{Pred}(\phi)} \texttt{sort}(P)$ represents the domains of the predicates in ϕ (or, more precisely, the sort symbols associated with these domains). We extend the notation to $\texttt{Dom}(\Phi) = \bigcup_{\phi \in \Phi} \texttt{Dom}(\phi)$ for Φ a set of formulae.

Principle 1 (Minimality)
A similarity measure $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}$ *satisfies* Minimality *iff for all* $A, B \in \texttt{Arg}(\texttt{OS}-\texttt{FOL})$, *if*
1. A and B are not trivial,
2. $\forall s_i \in \texttt{Dom}(\texttt{Supp}(A))$, $\nexists s_j \in \texttt{Dom}(\texttt{Supp}(B))$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
3. $\forall s_i \in \texttt{Dom}(\texttt{Conc}(A))$, $\nexists s_j \in \texttt{Dom}(\texttt{Conc}(B))$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
then $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, B) = 0$.

The first condition excludes the case where the arguments have no formula in the support and therefore no sort to compare and the second and third conditions ensure that each argument has completely different information.

The second (resp. third) principle states that the more an argument shares formulae in its support (resp. conclusion) with an another one, the higher is their similarity. For these principles, we need to introduce the notation $\mathbb{C}$ which represents the set of all grounded clauses in OS − FOL.

Principle 2 (Monotony – Strict Monotony)
A similarity measure $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}$ *satisfies* Monotony *iff for all* $A, B, C, A^*, B^*, C^* \in \texttt{Arg}(\texttt{OS}-\texttt{FOL})$, *if*
1. $\texttt{UC}(\texttt{Conc}(A)) = \texttt{UC}(\texttt{Conc}(B))$ *or* $\forall s_i \in \texttt{Dom}(\texttt{Conc}(A))$, $\nexists s_j \in \texttt{Dom}(\texttt{Conc}(C))$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
2. $\texttt{UC}(\texttt{Supp}(A)) \cap \texttt{UC}(\texttt{Supp}(C)) \subseteq \texttt{UC}(\texttt{Supp}(A)) \cap \texttt{UC}(\texttt{Supp}(B))$,
3. for $B_A = \texttt{UC}(\texttt{Supp}(B)) \setminus \texttt{UC}(\texttt{Supp}(A))$ *and* $C_A = \texttt{UC}(\texttt{Supp}(C)) \setminus \texttt{UC}(\texttt{Supp}(A))$, $B_A \subseteq C_A$, $C_A \setminus B_A \subseteq \mathbb{C}$ *and* $\forall s_i \in \texttt{Dom}(\texttt{Supp}(A))$, $\nexists s_j \in \texttt{Dom}(C_A \setminus B_A)$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
then $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, B) \geq \texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, C)$.
(Monotony)
– If the inclusion in condition 2. is strict or, $\texttt{UC}(\texttt{Supp}(A)) \cap \texttt{UC}(\texttt{Supp}(C)) \neq \emptyset$ *and* $B_A \subset C_A$,
then $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, B) > \texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, C)$.
(Strict Monotony)

Principle 3 (Dominance – Strict Dominance)
A similarity measure $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}$ *satisfies* Dominance *iff for all* $A, B, C, A^*, B^*, C^* \in \texttt{Arg}(\texttt{OS}-\texttt{FOL})$, *if*
1. $\texttt{UC}(\texttt{Supp}(B)) = \texttt{UC}(\texttt{Supp}(C))$,
2. $\texttt{UC}(\texttt{Conc}(A)) \cap \texttt{UC}(\texttt{Conc}(C)) \subseteq \texttt{UC}(\texttt{Conc}(A)) \cap \texttt{UC}(\texttt{Conc}(B))$,
3. for $B_A = \texttt{UC}(\texttt{Conc}(B)) \setminus \texttt{UC}(\texttt{Conc}(A))$ *and* $C_A = \texttt{UC}(\texttt{Conc}(C)) \setminus \texttt{UC}(\texttt{Conc}(A))$, $B_A \subseteq C_A$, $C_A \setminus B_A \subseteq \mathbb{C}$ *and* $\forall s_i \in \texttt{Dom}(\texttt{Conc}(A))$, $\nexists s_j \in \texttt{Dom}(C_A \setminus B_A)$ *s.t.* $s_i \prec s_j$ *or* $s_j \prec s_i$ *or* $s_j = s_i$,
then $\texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, B) \geq \texttt{sim}^{\texttt{Arg}(\texttt{OS}-\texttt{FOL})}(A, C)$.
(Dominance)

– *If the inclusion in cond. 2. is strict or,* $\mathtt{UC}(\mathtt{Conc}(A)) \cap \mathtt{UC}(\mathtt{Conc}(C)) \neq \emptyset$ *and* $B_A \subset C_A$*, then* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}(A, B) > \mathtt{sim}^{\mathtt{Arg(OS-FOL)}}(A, C)$.

(Strict Dominance)

Notice that we consider in the two last principles only arguments having no irrelevant information (i.e., $A^*, B^*, C^* \in \mathtt{Arg(OS-FOL)}$) allowing safe handling of their similarity. The first conditions allow to isolate the interesting behaviours on second and third conditions. Please note that the constraints $C_A \setminus B_A \subseteq \mathbb{C}$ ensure that the distinct elements in C cannot have similarity with A.

3 Similarity Models

To define the similarity between two arguments, we will split the reasoning in several steps, corresponding to the different levels used in the construction of the arguments. At each level, different similarity measures can be used to compare the objects, and various aggregation functions can then be used to go from the comparison of objects to the comparison of sets of objects (leading to the next level). This level structure is based on the fact that our arguments are built from CNF formulae. More precisely,

Level 1 : compute the similarity between two literals, by combining the similarity between their polarity, the predicate involved, and the predicates parameters (Section 3.1);

Level 2 : then we use the previous level and aggregate the result of comparing literals in order to compare grounded clauses (Section 3.2);

Level 3 : next, we aggregate the similarity between grounded clauses to obtain the similarity between sets of grounded clauses (Section 3.3);

Level 4 : finally, we can define the similarity between sets of instantiations, since each instantiation is a set of grounded clauses (Section 3.4).

The similarity between two arguments is obtained by computing the similarity between the instantiations of their supports and the similarity between their conclusions, so Level 4 is the last level of abstraction that we need.

3.1 Similarity between literals

Recall that a literal is a predicate with or without a negation operator "¬". To know how similar are two literals, we compute the similarity between two atoms (*i.e.* without the literals' polarity) and combine these scores according to the polarity. At the level of atoms, we identify two parameters influencing the similarity : the value of the predicates and those of their vectors of parameters. Thus the similarity between two atoms can be seen as a combination of three functions : c to compute the similarity between two vectors of constants, p between two predicates and g to aggregate these scores.

Definition 12 (Similarity between Atoms) *Let* $\mathbf{c} : \bigcup_{j,k=1}^{+\infty} \mathbf{C}^j \times \mathbf{C}^k \to [0,1]$ *be a similarity measure between a pair of vectors of constants,* $\mathbf{p} : \mathbf{P} \times \mathbf{P} \to [0,1]$ *be a similarity measure between a pair of predicates and* $\mathbf{g} : [0,1] \times [0,1] \to [0,1]$ *be an aggregation function. Given* $P_1, P_2 \in \mathbf{P}$ *with two vectors of constants* $A = \langle a_1, \dots, a_j \rangle$*,* $B = \langle b_1, \dots, b_k \rangle$ *where* $\forall a \in A, a \in \mathbf{C}$ *and* $\forall b \in B, b \in \mathbf{C}$*. To compute the similarity score between two atoms we define* $\mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle} : \bigcup_{j,k=1}^{+\infty} \mathbf{P} \times \mathbf{C}^j \times \mathbf{P} \times \mathbf{C}^k \to [0,1]$ *s.t.* $\mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}(P_1, A, P_2, B) = \mathbf{g}\Big(\mathbf{p}(P_1, P_2), \mathbf{c}\big(A, B\big)\Big)$.

A possible $\mathbf{p}$ is the function returning 1 if the predicates are the same, 0 otherwise.

Definition 13 (Function Equal) *Let* x, y *be two arbitrary objects. The function* $\mathtt{eq} : \mathbb{X} \times \mathbb{X} \to \{0, 1\}$ *is defined by* $\mathtt{eq}(x, y) = 1$ *if* $x = y$*; or* $\mathtt{eq}(x, y) = 0$ *otherwise.*

We propose an instance of $\mathbf{c}$ suited to vectors of objects.

Definition 14 (Pointwise Similarity) *Let* $X = \langle x_1, \dots, x_j \rangle, Y = \langle y_1, \dots, y_k \rangle$ *be arbitrary vectors of objects. The* pointwise similarity *between* X *and* Y *is :*

$$\mathtt{pws}(X, Y) = \begin{cases} 1 & X = Y = \emptyset \\ \frac{\sum_{i=1}^{\mathtt{min}(j,k)} \mathtt{eq}(x_i, y_i)}{\mathtt{max}(j,k)} & \textit{otherwise} \end{cases}$$

Having a similarity score between two atoms, we propose to use the polarities as binary factors of acceptance or not of the similarity between atoms.

Definition 15 (Similarity between Literals) *Let two literals* $l_1, l_2 \in \mathbf{L}$*. We define* $\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle} : \mathbf{L} \times \mathbf{L} \to [0,1]$*, the similarity measure between two literals according to a similarity measure between atoms* $\mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}$ *s.t. :* $\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}(l_1, l_2) =$

$$\begin{cases} \mathtt{simA}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}(\mathtt{Pred}(l_1), \mathtt{Para}(l_1), \\ \qquad \mathtt{Pred}(l_2), \mathtt{Para}(l_2)) \quad \textit{if } \mathtt{Pol}(l_1) = \mathtt{Pol}(l_2) \\ 0 \qquad \textit{otherwise} \end{cases}$$

3.2 Similarity between grounded clauses

From the level two of the definition of our similarity measures on arguments, we will need several mathematical tools that can be defined in an abstract way. In this part, we apply these tools only for level 2 (the comparison of two CNF formulae), but they will be applicable also at the next levels. Let us start with the notion of aggregation function.

Definition 16 (Aggregation Function) *Let* $\mathbb{X}$ *a set of objects and* $\{x_1, x_2, \dots\} \subseteq \mathbb{X}$ *some objects in this set. We say that* $\oplus$ *is an aggregation function if* $\forall k \in \mathbb{N}$*,* $\oplus$ *is a mapping* $[0,1]^k \to [0,1]$ *s.t.*

– *if* $x_i \geq x_i'$*, then* $\oplus(x_1, \dots, x_i, \dots, x_k) \geq$

$\oplus(x_1, \ldots, x'_i, \ldots, x_k)$ **(non-decreasingness)**
– $\oplus(0, \ldots, 0) = 0$ **(weak minimality)**
– $\forall i \in \{1, \ldots, k\}, \oplus(x_i) = x_i$ **(identity)**

These properties are satisfied by *e.g.* `min`, `max` and `avg`.

Now we introduce the notion of *membership* function which expresses how much an object is similar to the elements of a set.

Definition 17 (Membership Function) *Given $\mathbb{X}$ a set of objects, $x \in \mathbb{X}$ an object, $X \subseteq \mathbb{X}$, $\oplus$ an aggregation function and* `sim` *a similarity measure the* membership function *of x in X, $\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}} : \mathbb{X} \times 2^{\mathbb{X}} \to [0,1]$ is defined by : $\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(x, X) = \oplus_{x' \in X}(\mathtt{sim}^{\mathbb{X}}(x, x'))$.*

It is interesting to note that classical set-membership can be captured by $\varepsilon_{\mathtt{max},\mathtt{eq}}$ where `eq` is the equality function from Definition 13. Now we can evaluate how much a literal is similar to a clause, *i.e.* a set of literals : given $l \in \mathbf{L}$ a literal, $L \subseteq \mathbf{L}$ a set of literals and $\oplus^1$ an aggregation function, we define the function $\mathtt{s}^{\mathbf{L}} = \varepsilon^{\mathbf{L}}_{\oplus^1,\mathtt{simL}^{\langle \mathtt{g},\mathtt{p},\mathtt{c}\rangle}}$. Then, the similarity between two grounded clauses is computed by $\mathtt{simC}^{\mathtt{s}^{\mathbf{L}}}$.

Tversky [27] proposed the "ratio model", a general similarity measure which encompasses different well known similarity measure as the Jaccard measure [18], Dice measure [15], Sorensen one [26], Symmetric Anderberg [9] and Sokal and Sneath 2 [25]. We propose to extend it in two different ways. Firstly, instead of using the usual operators of membership of an element to a set, we propose to use our parameterisable membership function ε (see Definition 17). Then a new parameter γ is added allowing us to vary these scores in an increasing or decreasing way only in the cases where the sets of objects are partially similar.

Definition 18 (Extended Tversky Measure) *Let $X, Y \subseteq \mathbb{X}$ be arbitrary sets of objects. Let $\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}$ be a membership function with $\oplus$ an aggregation function and* `sim` *a similarity measure. We denote by* `avg` *the average function. Let us consider $a = \mathtt{avg}\Big(\sum_{x \in X} \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(x, Y), \sum_{y \in Y} \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(y, X)\Big)$, $b = \sum_{x \in X} 1 - \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(x, Y)$, $c = \sum_{y \in Y} 1 - \varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}(y, X)$, and $\alpha, \beta \in [0, +\infty[$, $\gamma \in]0, +\infty[$. The* extended Tversky measure *between X and Y is :*

$$\mathtt{Tve}^{\alpha,\beta,\gamma,\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}}(X, Y) = \begin{cases} 1 & \text{if } X = Y = \emptyset \\ \left(\frac{a}{a + \alpha \cdot b + \beta \cdot c}\right)^{\gamma} & \text{otherwise} \end{cases}$$

Classical similarity measures (see Table 1 in [4] for the definitions) can be obtained with $\alpha = \beta = 2^{-n}$ and the classical set-membership. In particular, the Jaccard measure (i.e. `jac`) is obtained with $n = 0$, Dice (i.e. `dic`) with $n = 1$, Sorensen (i.e. `sor`) with $n = 2$, Anderberg (i.e. `adb`) with $n = 3$, and Sokal and Sneah 2 (i.e. $\mathtt{ss}_2$) with $n = -1$.

Under some reasonable assumptions, Tversky measure s.t. $\alpha = \beta$ are symmetric.

Symmetric Measures	Non-Symmetric Measures
$\mathtt{Tve}^{1,1,\otimes}(X,Y) = \mathtt{jac}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,1,\otimes}(X,Y) = \mathtt{ns\text{-}jac}^{\ominus}(X,Y)$
$\mathtt{Tve}^{0.5,0.5,\otimes}(X,Y) = \mathtt{dic}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,0.5,\otimes}(X,Y) = \mathtt{ns\text{-}dic}^{\ominus}(X,Y)$
$\mathtt{Tve}^{0.25,0.25,\otimes}(X,Y) = \mathtt{sor}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,0.25,\otimes}(X,Y) = \mathtt{ns\text{-}sor}^{\ominus}(X,Y)$
$\mathtt{Tve}^{0.125,0.125,\otimes}(X,Y) = \mathtt{adb}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,0.125,\otimes}(X,Y) = \mathtt{ns\text{-}adb}^{\ominus}(X,Y)$
$\mathtt{Tve}^{2,2,\otimes}(X,Y) = \mathtt{ss_2}^{\ominus}(X,Y)$	$\mathtt{Tve}^{0,2,\otimes}(X,Y) = \mathtt{ns\text{-}ss_2}^{\ominus}(X,Y)$

Table 1 – Set of parametric (non-)symmetric measures, where $\otimes$ is $\gamma, \varepsilon^{\mathbb{X}}_{\mathtt{max},\mathtt{sim}}$ and $\ominus$ is $\gamma, \mathtt{sim}^{\mathbb{X}}$

Proposition 1 *For any $X, Y \subseteq \mathbb{X}$, any $\gamma \in]0, +\infty[$, any membership function $\varepsilon^{\mathbb{X}}_{max,\mathtt{sim}}$ s.t.* `sim` *is symmetric, we have $\mathtt{Tve}^{\alpha,\alpha,\otimes}(X, Y) = \mathtt{Tve}^{\alpha,\alpha,\ominus}(Y, X)$, where $\otimes = \gamma, \varepsilon^{\mathbb{X}}_{max,\mathtt{sim}}$ and $\ominus = \gamma, \varepsilon^{\mathbb{X}}_{max,\mathtt{sim}}$.*

In the rest of the paper we will focus our study on membership function using the aggregator function `max`. Table 1 denotes the set of parametric (non-)symmetric extended versions of the well known similarity measures, where fixing α and β corresponds to choosing among Jaccard, Dice, Sorensen, Anderberg, or Sokal and Sneah.

The other parameters of the different similarity measures are only the coefficient γ and the similarity function $\mathtt{sim}^{\mathbb{X}}$. Please note that γ allows us to have a lower evaluation between a set of literals than a set of clauses (or instantiations), i.e. when sets of objects are interpreted disjunctively or conjunctively. Let us prove that any such measure satisfies some intuitive properties : two sets are maximally similar if they are identical (in the symmetric case), or at least included in one another (non-symmetric case).

Proposition 2 *If $\mathtt{sim}^{\mathbb{X}}$ satisfies Maximality [4], then, for any $\gamma \in]0, +\infty[$, $\alpha \neq 0$, if*
– $Y = X$ then $\mathtt{Tve}^{\alpha,\alpha,\ominus}(X, Y) = 1$ (symmetric case),
– $Y \subseteq X$ then $\mathtt{Tve}^{0,\alpha,\ominus}(X, Y) = 1$ (non-symmetric case).

Example 6 *Let $P_1 = P(A, B)$, $P_2 = P(A, C)$ and $P_3 = P(C, B)$. Let $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}$. $\mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}}(P_1, P_2 \vee P_3) = \mathtt{Tve}^{1,1,1,\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}}(P_1, P_2 \vee P_3) = \frac{a}{a+b+c} = \frac{1}{3}$ with :*

- *$a = \mathtt{avg}(\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2 \vee P_3), \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_2, P_1) + \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_3, P_1)) = \mathtt{avg}(\frac{1}{2}, 1) = \frac{3}{4}$,*
- *$b = 1 - \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2 \vee P_3) = \frac{1}{2}$,*
- *$c = (1 - \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_2, P_1)) + (1 - \varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_3, P_1)) = \frac{1}{2} + \frac{1}{2} = 1$, with $\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2 \vee P_3) = \frac{1}{2} = \mathtt{max}(\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P_1, P_2), \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P_1, P_3))$, $\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_2) = \mathtt{max}(\mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws}\rangle}(P_1, P_2)) = \frac{1}{2}$ (idem for $\varepsilon^{\mathbf{L}}_{\mathtt{max},\mathtt{s}^{\mathbf{L}}}(P_1, P_3)$).*

3.3 Similarity between grounded clauses

We introduce $\mathbb{C}$ the set of all grounded clauses in `OS−FOL`.

Definition 19 (Grounded clause membership) *Let $\delta \in \mathbb{C}$ be a grounded clause and $\Delta \subseteq \mathbb{C}$ be a set of grounded clauses. Let an aggregation function $\oplus^{\mathtt{c}}$ and a similarity measure between a pair of clauses $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}$, with $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}$. The membership function of a grounded clause in a set of grounded clauses, denoted $\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}} : \mathbb{C} \times 2^{\mathbb{C}} \to [0,1]$, is $\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}(\delta, \Delta) = \oplus^{\mathtt{c}}_{\delta' \in \Delta}(\mathtt{s}^{\mathbb{C}}(\delta, \delta'))$.*

Definition 20 (Similarity between sets of grounded clauses) *Let $\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}$ be a membership function with $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}$ and $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}$. A similarity measure between two sets of grounded clauses is defined as $\mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}} : 2^{\mathbb{C}} \times 2^{\mathbb{C}} \to [0,1]$.*

3.4 Similarity between instantiations

Now, define $\mathbb{I}$ the set of all instantiations in $\mathtt{OS-FOL}$.

Definition 21 (Instantiation membership) *Let an instantiation $\Delta \in \mathbb{I}$ and a set of instantiations $I \subseteq \mathbb{I}$. Let an aggregation function $\oplus^{\mathtt{i}}$ and a similarity measure between a pair of set of clauses $\mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}$ with $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}$ and $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}$. The membership function of an instantiation in a set of instantiations, $\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}} : \mathbb{I} \times 2^{\mathbb{I}} \to [0,1]$, is $\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}(\Delta, I) = \oplus^{\mathtt{i}}_{\Delta' \in I}(\mathtt{s}^{\mathbb{I}}(\Delta, \Delta'))$.*

Definition 22 (Similarity between sets of instantiations) *Let $\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}$ be a membership function with $\mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}$, $\mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}$ and $\mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}$. The similarity measure between two set of instantiations is defined as $\mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}} : 2^{\mathbb{I}} \times 2^{\mathbb{I}} \to [0,1]$.*

Let define a similarity measure between sets of formulae.

Definition 23 (Similarity Models) *A Similarity Model ($\mathtt{SM}$) is a tuple $\mathbf{M} = \langle \mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}, \mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}, \mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}, \mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}} \rangle$. Let two sets of formulae $\Phi, \Psi \subseteq \mathtt{OS-FOL}$ and $\mathbf{I_{St}}$ an interpretation over a structure $\mathbf{St}$. The similarity between Φ and Ψ is $\mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M},\mathbf{I_{St}}}(\Phi, \Psi) = \mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}}(\mathtt{Inst}_{\mathbf{I_{St}}}(\Phi), \mathtt{Inst}_{\mathbf{I_{St}}}(\Psi))$.*

Finally, using the measure of similarity between sets of formulae, we can extend the definition from [4] to asses the similarity between two $\mathtt{OS-FOL}$ arguments.

Definition 24 (Similarity between OS-FOL Arguments) *Let a coefficient $0 < \eta < 1$, a $\mathtt{SM}$ $\mathbf{M}$ and $\mathbf{I_{St}}$ an interpretation over a structure $\mathbf{St}$. We define $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta} : \mathtt{Arg(OS-FOL)} \times \mathtt{Arg(OS-FOL)} \to [0,1]$ by $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}(A,B) = \eta \cdot \mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M},\mathbf{I_{St}}}(\mathtt{UC}(\mathtt{Supp}(A)), \mathtt{UC}(\mathtt{Supp}(B)))$ $+(1-\eta) \cdot \mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M},\mathbf{I_{St}}}(\mathtt{UC}(\mathtt{Conc}(A)), \mathtt{UC}(\mathtt{Conc}(B)))$.*

Example 7 *Let $\mathbf{M}_{\mathtt{jac}} = \langle \mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathtt{min},\mathtt{eq},\mathtt{pws} \rangle}, \mathtt{s}^{\mathbb{C}} = \mathtt{jac}^{2,\mathtt{s}^{\mathbf{L}}}, \mathtt{s}^{\mathbb{I}} = \mathtt{jac}^{1,\mathtt{s}^{\mathbb{C}}}, \mathtt{jac}^{1,\mathtt{s}^{\mathbb{I}}} \rangle$ be a similarity instantiation model and let A_1 and A_2 be the two OS-FOL arguments from Example 5. Their respective instantiations are given in Example 4 for the premises and the conclusions. Let us compute the similarity between A_1 and A_2 with $\eta = 0.5$.* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}},0.5}(A_1, A_2) =$ $0.5 \cdot \mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Supp}(A_1), \mathtt{Supp}(A_2)) +$ $0.5 \cdot \mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Conc}(A_1), \mathtt{Conc}(A_2))$ $= 0.5 \cdot \frac{73}{1143} + 0.5 \cdot \frac{5}{11} \simeq 0.2592$ *where* $\mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Supp}(A_1), \mathtt{Supp}(A_2)) =$ $\mathtt{jac}^{1,\mathtt{s}^{\mathbb{I}}}(\mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Supp}(A_1)), \mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Supp}(A_2))) = \frac{73}{1143} \simeq 0.064$ *and* $\mathtt{sim}^{\mathtt{OS-FOL}}_{\mathbf{M}_{\mathtt{jac}},\mathbf{I_{St}}}(\mathtt{Conc}(A_1), \mathtt{Conc}(A_2)) =$ $\mathtt{jac}^{1,\mathtt{s}^{\mathbb{I}}}(\mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Conc}(A_1)), \mathtt{Inst}_{\mathbf{I_{St}}}(\mathtt{Conc}(A_2))) = \frac{5}{11} \simeq 0.4545$.

4 Axiomatic Evaluation

Before determining the principles satisfied by our similarity measures, we introduce the notion of well-behaved $\mathtt{SM}$. It is a bridge between the (lower level) properties of the measures that we use (*e.g.* the Tversky measures) and the (higher level) properties of the similarity measure between arguments defined from such a $\mathtt{SM}$.

Definition 25 (Well-Behaved $\mathtt{SM}$) *A $\mathtt{SM}$ $\mathbf{M} = \langle \mathtt{s}^{\mathbf{L}} = \mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}, \mathtt{s}^{\mathbb{C}} = \mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}, \mathtt{s}^{\mathbb{I}} = \mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}},\mathtt{s}^{\mathbb{C}}}}, \mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}},\mathtt{s}^{\mathbb{I}}}} \rangle$ is well-behaved iff :*

1. (a) i. $\mathbf{g}(1,1) = 1$,
   ii. $\mathbf{g}(0,0) = 0$,
   (b) i. $\mathbf{p}(P,P) = 1$,
   ii. $\mathbf{p}(P,Q) = 0$ *iff* $P \neq Q$,
   (c) i. $\mathbf{c}(\langle a_1, \ldots, a_k \rangle, \langle a_1, \ldots, a_k \rangle) = 1$,
   ii. *if* $\forall i \in \{1, \ldots, k\}$, $\nexists j \in \{1, \ldots, n\}$ *s.t.* $a_i = b_j$ *then* $\mathbf{c}(\langle a_1, \ldots, a_k \rangle, \langle b_1, \ldots, b_n \rangle) = 0$,
2. *Given $\mathbb{X}$ a set of objects,*
   (a) $\mathtt{sim}^{\varepsilon,\mathtt{s}}(X,X) = 1$ *for any set of objects $X \subseteq \mathbb{X}$,*
   (b) *if $\forall x \in X$, $\forall x' \in X'$, $\mathtt{s}(x,x') = 0$ then $\mathtt{sim}^{\varepsilon,\mathtt{s}}(X,X') = 0$,*
   (c) *let $X_0, X_1, X_2 \subseteq \mathbb{X}$ s.t. $X_1 \subset X_2$ and $X_2 \setminus X_1 = \{x_2\}$. If $\exists x_0 \in X_0$ s.t. $\mathtt{s}(x_0,x_2) = \mathtt{s}(x_2,x_0) = 1$ then $\mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0,X_2) \geq \mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0,X_1)$,*
   (d) *let $X_0, X_1, X_2 \subseteq \mathbb{X}$ s.t. $X_1 \subset X_2$ and $X_2 \setminus X_1 = \{x_2\}$. If $\forall x_0 \in X_0$, $\mathtt{s}(x_0,x_2) = \mathtt{s}(x_2,x_0) = 0$ then $\mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0,X_1) \geq \mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0,X_2)$.*

In the last item, $\mathbb{X}$ can be the set of all literals (for characterizing $\mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{1},\mathtt{s}^{\mathbf{L}}}}$), the set of all grounded clauses (for

Table 2 – Satisfaction of the principles of similarity measures. The symbol • (resp. ∘) means the measure satisfies (resp. violates) the principle. $\mathtt{sim}_x$ is a shorthand for $\mathtt{sim}_x^{\mathtt{Arg(OS-FOL)}}$.

	$\mathtt{sim}_{\mathtt{jac}}$	$\mathtt{sim}_{\mathtt{dic}}$	$\mathtt{sim}_{\mathtt{sor}}$	$\mathtt{sim}_{\mathtt{adb}}$	$\mathtt{sim}_{\mathtt{ss_2}}$	$\mathtt{sim}_{\mathtt{ns}}$-jac	$\mathtt{sim}_{\mathtt{ns}}$-dic	$\mathtt{sim}_{\mathtt{ns}}$-sor	$\mathtt{sim}_{\mathtt{ns}}$-adb	$\mathtt{sim}_{\mathtt{ns}}$-ss$_2$
Maximality	•	•	•	•	•	•	•	•	•	•
Symmetry	•	•	•	•	•	∘	∘	∘	∘	∘
Substitution	•	•	•	•	•	∘	∘	∘	∘	∘
Syntax Independence	•	•	•	•	•	•	•	•	•	•
Minimality	•	•	•	•	•	•	•	•	•	•
Monotony	•	•	•	•	•	•	•	•	•	•
Strict Monotony	•	•	•	•	•	∘	∘	∘	∘	∘
Dominance	•	•	•	•	•	•	•	•	•	•
Strict Dominance	•	•	•	•	•	∘	∘	∘	∘	∘

characterizing $\mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}}, \mathtt{s}^{\mathbb{C}}}}$) or the set of instantiations (for characterizing $\mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}}, \mathtt{s}^{\mathbb{I}}}}$). Now we can show that a well-behaved SM guarantees that the corresponding similarity measure satisfies some principles.

Theorem 1 *For any* $\mathbf{M} \in \mathtt{SM}$*, if* $\mathbf{M}$ *is well-behaved then* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}$ *satisfies the following principles : Maximality, Minimality, Monotony and Dominance.*

To satisfy other principles we propose additional constraints.

Theorem 2 *Let a well-behaved* $\mathbf{M} \in \mathtt{SM}$ *and* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}$ *a similarity based on* $\mathbf{M}$.
– $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}$ *satisfies Symmetry (resp. Syntax Independence) if all the functions in* $\mathbf{M}$ *are symmetric (resp. syntax independent).*
– $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}$ *satisfies Strict Monotony and Strict Dominance if it satisfies condition 2.(c') : let* $X_0, X_1, X_2 \subseteq \mathbb{X}$ *s.t.* $X_1 \subset X_2$ *and* $X_2 \setminus X_1 = \{x_2\}$*. If* $\mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0, X_1) < 1$ *and* $\exists x_0 \in X_0$ *s.t.* $\mathtt{s}(x_0, x_2) = \mathtt{s}(x_2, x_0) = 1$ *then* $\mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0, X_2) > \mathtt{sim}^{\varepsilon,\mathtt{s}}(X_0, X_1)$.

We extend some results from [4].

Proposition 3 *Let* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}$ *a similarity measure.*
– *Let* $A, B \in \mathtt{Arg(OS-FOL)}$*, if* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}$ *satisfies Maximality, Monotony, Strict Monotony and Strict Dominance then* $A \approx B$ *iff* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}(A, B) = 1$.
– *If* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}$ *satisfies Symmetry, Maximality, Strict Monotony, Dominance, and Strict Dominance then* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}$ *satisfies Substitution.*

Let us prove that the functions **g**, **p** and **c** used in the paper satisfy the expected properties of a well-behaved SM.

Lemma 1 *For* $\mathbf{g} \in \{\mathtt{min}, \mathtt{avg}\}$*,* $\mathbf{p} = \mathtt{eq}$ *and* $\mathbf{c} = \mathtt{pws}$*,* $\langle \mathbf{g}, \mathbf{p}, \mathbf{c} \rangle$ *satisfies item 1. of Def. 25.*

We can show similar results for the Tversky measures that we use to define $\mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}}, \mathtt{s}^{\mathbf{L}}}}$, $\mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}}, \mathtt{s}^{\mathbb{C}}}}$ and $\mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}}, \mathtt{s}^{\mathbb{I}}}}$. We consider the measures described in Table 1.

Lemma 2 *If* $\mathtt{Tve}^{\alpha,\beta,\gamma,\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}}$ *is a Tversky measure, with* $\oplus = \mathtt{max}$*, and* $\mathtt{sim}$ *is*
– *either* $\mathtt{simL}^{\langle \mathbf{g},\mathbf{p},\mathbf{c} \rangle}$ *(from Definition 15) s.t.* $\langle \mathbf{g}, \mathbf{p}, \mathbf{c} \rangle$ *satisfies item 1. of Def. 25,*
– *or a similarity measure satisfying the item 2. of Def. 25, then* $\mathtt{Tve}^{\alpha,\beta,\gamma,\varepsilon^{\mathbb{X}}_{\oplus,\mathtt{sim}}}$ *satisfies the item 2. of Def. 25.*

Proposition 4 *For* $x \in \{\mathtt{jac}, \mathtt{dic}, \mathtt{sor}, \mathtt{adb}, \mathtt{ss_2}, \mathtt{ns\text{-}jac}, \mathtt{ns\text{-}dic}, \mathtt{ns\text{-}sor}, \mathtt{ns\text{-}adb}, \mathtt{ns\text{-}ss_2}\}$*, define* $\mathtt{sim}_x^{\mathtt{Arg(OS-FOL)}}$*. Then define the similarity model* $\mathtt{SM}$ $\mathbf{M}_x = \langle \mathtt{simL}^{\langle \mathtt{min,eq,pws} \rangle}, x^{2,\mathtt{sim}^{\mathbf{L}}}, x^{1,\mathtt{sim}^{\mathbb{C}}}, x^{1,\mathtt{sim}^{\mathbb{I}}} \rangle$*. The satisfaction of principles by the measures is given in Table* 2*.*

Notice that Proposition 4 implies that all the principles are compatible. Moreover with the result of item 1 of Proposition 3, we can deduce that our 5 symmetric extended Tversky measures satisfying a stronger form of maximality, since equivalent arguments are maximally similar. For non-symmetric measures, we show that they can obtain full similarity in a particular case of sub-argument.

Proposition 5 *Let* $A, B \in \mathtt{Arg(OS-FOL)}$ *be two arguments. Assume that* $\mathbf{M}$ *is a* SM *s.t.* $\mathtt{simC}^{\varepsilon^{\mathbf{L}}_{\oplus^{\mathtt{l}}, \mathtt{s}^{\mathbf{L}}}}$*,* $\mathtt{simI}^{\varepsilon^{\mathbb{C}}_{\oplus^{\mathtt{c}}, \mathtt{s}^{\mathbb{C}}}}$ *and* $\mathtt{simSI}^{\varepsilon^{\mathbb{I}}_{\oplus^{\mathtt{i}}, \mathtt{s}^{\mathbb{I}}}}$ *are Tversky measures s.t.* $\alpha \neq \beta$ *for at least one of them (*i.e. *it is non-symmetric). If* B *is a sub-argument of* A*, then* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}(A, B) \geq \eta$*. Moreover, if* $\mathtt{UC}(\mathtt{Conc}(B)) \subseteq \mathtt{UC}(\mathtt{Conc}(A))$*, then* $\mathtt{sim}^{\mathtt{Arg(OS-FOL)}}_{\mathbf{M},\mathbf{I_{St}},\eta}(A, B) = 1$.

5 Conclusion

In this paper, we have proposed the rich methodology of similarity models which are able to express large families of similarity measures between Order-Sorted First Order Logic (OS − FOL) arguments, thanks to various parameters which allow to define generalized versions of similarity measures from the literature. For the first time in the logical argumentation literature, we define non-symmetric similarity measures. A set of nine principles for these OS − FOL arguments has been proposed with a set of well-behaved properties ensuring some principles. We have shown that our symmetric measures satisfy all the principles, while their non-symmetric counterparts only satisfy a subset.

This work paves the way to several interesting research questions. First of all, we can consider additional measures (*e.g.* Ochiai [24], Kulczynski [20]) and principles (*e.g.* triangular inequality, non-zero, independent distribution [14]) to allow a more accurate comparison of similarity measures. Another research line could be to consider situations where different predicates are partially similar. For instance, one can consider that $greaterOrEqual(A, B)$ is somehow similar to $strictlyGreater(A, B)$. Following the same idea as in [6], we also plan to use our similarity measures as a parameter of acceptability semantics. Finally, we want to apply our work on real data expressed in fragments of $\mathtt{OS-FOL}$.

Références

[1] Amgoud, L.: *A Replication Study of Semantics in Argumentation*. Dans *IJCAI'19*, pages 6260–6266, 2019.

[2] Amgoud, L. et J. Ben-Naim: *Axiomatic Foundations of Acceptability Semantics*. Dans *KR'16*, pages 2–11, 2016.

[3] Amgoud, L., E. Bonzon, J. Delobelle, D. Doder, S. Konieczny et N. Maudet: *Gradual Semantics Accounting for Similarity between Arguments*. Dans *KR'18*, pages 88–97, 2018.

[4] Amgoud, L. et V. David: *Measuring Similarity between Logical Arguments*. Dans *KR'18*, pages 98–107, 2018.

[5] Amgoud, L. et V. David: *An Adjustment Function for Dealing with Similarities*. Dans *COMMA'20*, pages 79–90, 2020.

[6] Amgoud, L. et V. David: *A General Setting for Gradual Semantics Dealing with Similarity*. Dans *AAAI'21*, 2021.

[7] Amgoud, L. et V. David: *Similarity Measures Based on Compiled Arguments*. Dans *ECSQARU'21*, pages 32–44, 2021.

[8] Amgoud, L., V. David et D. Doder: *Similarity Measures Between Arguments Revisited*. Dans *ECSQARU'19*, pages 3–13, 2019.

[9] Anderberg, M.: *Cluster analysis for applications. Monographs and textbooks on probability and mathematical statistics*, 1973.

[10] Arioua, A., M. Croitoru et S. Vesic: *Logic-based argumentation with existential rules*. Int. J. Approx. Reason., 90 :76–106, 2017.

[11] Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi et P. F. Patel-Schneider (rédacteurs): *The Description Logic Handbook : Theory, Implementation, Applications*. 2003.

[12] Besnard, P. et A. Hunter: *A logic-based theory of deductive arguments*. Artificial Intelligence, 128(1-2) :203–235, 2001.

[13] Besnard, P. et A. Hunter: *Practical first-order argumentation*. Dans *AAAI'05*, pages 590–595, 2005.

[14] David, V.: *Dealing with Similarity in Argumentation*. Thèse de doctorat, Univ. Toulouse III, 2021.

[15] Dice, L.: *Measures of the amount of ecologic association between species*. Ecology, 26(3) :297–302, 1945.

[16] Governatori, G., M. Maher, G. Antoniou et D. Billington: *Argumentation Semantics for Defeasible Logic*. J. Log. Comput., 14(5) :675–702, 2004.

[17] Halpern, J. et V. Weissman: *Using first-order logic to reason about policies*. TISSEC, 11(4) :1–41, 2008.

[18] Jaccard, P.: *Nouvelles recherches sur la distributions florale*. Bulletin de la societe Vaudoise des sciences naturelles, 37 :223–270, 1901.

[19] Jantke, K. P.: *Nonstandard Concepts of Similarity in Case-Based Reasoning*. Dans *Information Systems in Data Analysis : Prospects – Foundations – Applications*, pages 28–43, 1994.

[20] Kulczynski, S.: *Classe des sciences mathématiques et naturelles*. Bull. Internat. de l'Academie Polonaise des Sciences et des Lettres, pages 57–203, 1927.

[21] Lang, J., P. Liberatore et P. Marquis: *Propositional independence-formula-variable independence and forgetting*. J. Artif. Intell. Res., 18 :391–443, 2003.

[22] Longo, C., F. Longo et C. Santoro: *CASPAR : Towards decision making helpers agents for IoT, based on natural language and first order logic reasoning*. Eng. Appl. Artif. Intell., 104 :104269, 2021.

[23] McBurney, Peter, Simon Parsons et Iyad Rahwan (rédacteurs): *Proc. ArgMAS'11*, 2012.

[24] Ochiai, A.: *Zoogeographical studies on the Soleoid fishes found in Japan and its neighbouring regions*. Bull. Jpn. Soc. scient. Fish., 22 :526–530, 1957.

[25] Sneath, P. et R. Sokal: *Numerical taxonomy. The principles and practice of numerical classification.* 1973.

[26] Sørensen, T.: *A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons*. Biol. Skr., 5 :1–34, 1948.

[27] Tversky, A.: *Features of Similarity*. Psychological Review, 84(4) :327–352, 1977.

[28] Zhong, Q., X. Fan, X. Luo et F. Toni: *An explainable multi-attribute decision model based on argumentation*. Expert Sys. and Appl., 117 :42–61, 2019.