Reasoning with Inconsistent Possibilistic Description Logics Ontologies with Disjunctive Assertions*

Sergio Alejandro Gómez
Artificial Intelligence Research and Development Laboratory (LIDIA)
Department of Computer Science and Engineering - Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina
sag@cs.uns.edu.ar

ABSTRACT

We present a preliminary framework for reasoning with *possibilistic description logics ontologies with disjunctive assertions* (PoDLoDA ontologies for short). Given a PoDLoDA ontology, its terminological box is expressed in the description logic programming fragment but its assertional box allows four kinds of statements: an individual is a member of a concept, two individuals are related through a role, *an individual is a member of the union of two or more concepts* or *two individuals are related through the union of two or more roles*. Axioms and statements in PoDLoDA ontologies have a numerical certainty degree attached. A disjunctive assertion expresses a doubt respect to the membership of either individuals to union of concepts or pairs of individuals to the union of roles. Because PoDLoDA ontologies allow to represent incomplete and potentially inconsistent information, instance checking is addressed through an adaptation of Bodanza's Suppositional Argumentation System that allows to reason with modus ponens and constructive dilemmas. We think that our approach will be of use for implementers of reasoning systems in the Semantic Web where uncertainty of membership of individuals to concepts or roles is present.

Keywords: Suppositional argumentation, ontology reasoning, inconsistency handling, Possibilistic Description Logics, Semantic Web, Artificial Intelligence

1. INTRODUCTION

Reasoning with Description Logics (DL) [3] is fundamental in the ontology layer of the Semantic Web [8], where the meaning of data resources described in terms of the Web Ontology Language (OWL-DL) are machine processable. Traditionally, an ontology is a set of axioms that define intensionally (by means of a Tbox or terminological box) a set of concepts and roles and extensionally a set individuals belonging to some of the concepts and/or relating to each other through roles (by means of an Abox or assertional box).

In the last years approaches for reasoning with inconsistent ontologies have gained interest in the Semantic Web community. An inconsistent ontology implies that at least one individual is a member of both a concept and its complement, thus making anything derivable from such an ontology. Inconsistent ontologies pose a problem because they have to be debugged by the knowledge engineer prior to deployment of applications. Many times though it is not possible to do this because the domain being modeled is intrinsically contradictory or, as is the case with imported ontologies, the programmer does not have the authority to edit the ontology's contents. Many approaches have been proposed for dealing with this situation which is usually solved in one of two ways. The first kind of approaches consists of eliminating part of the ontology to restore consistency (e.g. approaches based on Belief Revision theory [24, 25] or on paraconsistent logics [22]). The second kind consists of accepting inconsistency and using a non-standard reasoning mechanism to get some meaningful answers from an inconsistent knowledge base (e.g. argumentation [2, 17, 18, 19]). In this paper we will rely on the latter approach.

Defeasible argumentation is an approach to non-monotonic reasoning that can be used when several diverging opinions may exist (each opinion is supported by an argument instead of a proof), so differences can be resolved by considering all pros and cons of a given claim through a dialectical analysis [4, 12, 27]. Given a claim, this process usually takes into account defeaters (i.e. other claims that are against the original claim and seem to have greater importance) and defeaters of those defeaters (thus maybe reinstating the original claim) as part of a recursive process that ends when no defeater can be found. Defeasible Logic Programming (DeLP) [14] is a rule-based implementation of defeasible argumentation based on the Prolog programming language. Possibilistic DeLP (or just PDeLP) [1] is another argumentative approach to non-monotonic reasoning that derives from DeLP. Unlike DeLP, that uses specificity for comparing arguments during the dialectical analysis, PDeLP uses possibilistic degrees associated to each rule to qualify arguments indicating its relative weight (or certainty), and thus imposing a rule priority principle for comparing arguments based on the idea that an argument is as credible as its weakest link.

The hypothesis underlying this work is that defeasible argumentation is a reliable tool for dealing with the problem of reasoning with possibly inconsistent ontologies having disjunctive assertions. Therefore, we will extend PDeLP for it to be able to handle facts with disjunctions. We call the extension presented here *Suppositional Possibilistic Defeasible Logic Programming* (SPDeLP). Therefore SPDeLP, beside being an extension of PDeLP, can be considered also an adaptation of Bodanza's Suppositional

*This paper is an extended and revised version of the article S.A. Gómez. A Preliminary Framework for Reasoning with Inconsistent Possibilistic Description Logics Ontologies with Disjunctive Assertions. *XXI Argentinian Conference of Computer Science (CACIC 2015)*, Junin, Buenos Aires, Argentina, 5–9 October 2015, 2015.

Argumentation System [10] with an emphasis in the features that will allow its future computational implementation.

We then extend traditional DL ontologies with disjunctive assertional statements of the form *"an individual is a member of the union of two or more classes"* and *"two individuals are related through the union of two or more roles"*. Each axiom in the ontology is coupled with a weight or certainty degree in order to decide between contradicting axioms about the membership of individuals to concepts, a reasoning task known as instance checking. We chose to call this new kind of ontologies *PoDLoDA ontologies*. We then interpret PoDLoDA ontologies as SPDeLP programs to answer queries of membership of individual to classes (or the relationship of pairs through roles) in the presence of inconsistency. We think that this approach could extend the range of applications in which ontologies are used, such as performing decision-making with uncertain knowledge. As proof of concept, we present a running example that shows how SPDeLP allows to compute instance checking in a small PoDLoDA ontology.

The rest of the article is structured as follows. In Section 2, we present the fundamentals of PoDLoDA ontologies. In Section 3, we describe the argumentation approach to reasoning with SPDeLP and how it is used to perform instance checking in PoDLoDA ontologies. In Section 4, we discuss related work. Finally, Section 5 concludes.

2. POSSIBILISTIC DESCRIPTION LOGIC ONTOLOGIES WITH DISJUNCTIVE ASSERTIONS

We introduce here possibilistic description logic ontologies with disjunctive assertions. In brief, they are ontologies with numeric degrees attached to axioms and that feature disjunctive assertions of membership of individuals to concepts and roles. We briefly recall reasoning in description logics along with the variation proposed for possibilistic description logics, and finally present ontologies with disjunctive assertions.

A Brief Recall of Description Logics

Description Logics (DL) are a well-known family of knowledge representation formalisms [3]. They are based on the notions of *concepts* (unary predicates, classes) and *roles* (binary relations), and are mainly characterized by the constructors that allow complex concepts and roles to be built from atomic ones. Let C and D stand for concepts and R for a role name. Concept descriptions are built from concept names using the constructors conjunction ($C \sqcap D$), disjunction ($C \sqcup D$), negation ($\neg C$), existential restriction ($\exists R.C$), and value restriction ($\forall R.C$). To define the semantics of concept descriptions, concepts are interpreted as subsets of a domain of interest, and roles as binary relations over this domain. Other constructors include inverse R^- and transitive R^+ roles.

A DL ontology consists of two finite and mutually disjoint sets: a *Tbox* which introduces the *terminology* and an *Abox* (assertional box) which contains facts about particular objects in the application domain. A Tbox contains inclusion axioms $C \sqsubseteq D$, where C and D are (possibly complex) concept descriptions, meaning that every individual of C is also a D, and equality axioms $C \equiv D$ meaning that C and D are equivalent concepts (i.e. every individual in C is an individual in D and vice versa). Objects in the Abox are referred to by a finite number of *individual names* and these names may be used in assertional statements: *concept assertions* of two types: $a : C$ (meaning the individual a is a member of concept C), and *role assertions* of the type $\langle a, b\rangle : R$ (meaning that a is related to b through the role R).

Many reasoning Tbox and Abox reasoning tasks are defined for DL ontologies (see [3]), but in this work we are only interested in *instance checking* that refers to determining if an individual is a member of a certain class.

One form of assigning semantics to a DL ontology is based on the fact that DL is isomorphic with first-order logic restricted to two variables. Then, for example, the inclusion axiom $C \sqsubseteq D$ can be interpreted as the first-order logic formula $(\forall x)(c(x) \rightarrow d(x))$ and an assertion $a : C$ as $c(a)$. Description Logic Programming (DLP) approaches [20] take advantage of this to interpret such axioms as the Prolog rules "d(X) :- c(X)." and "c(a).", resp. We will apply this in Section 3 to redefine instance checking in terms of defeasible argumentation allowing to infer that a is a member of the concept D by finding a proof for the goal ":- d(a)" (see [17] for details).

Fundamentals of Possibilistic Description Logics

We now recall the fundamentals of possibilistic description logic ontologies. Our presentation is based on [16] and [7]. Let $\mathcal{L}_{DL}$ be a DL description language, a *possibilistic DL ontology* is a set of possibilistic axioms of the form $(\varphi, W(\varphi))$ where φ is an axiom expressed in $\mathcal{L}_{DL}$ and $W(\varphi) \in [0, 1]$ is the degree of certainty (or priority) of φ. Namely, a possibilistic DL ontology Σ is such that $\Sigma = \{(\varphi_i, W(\varphi_i)) : i = 1, \ldots, n\}$. Only somewhat certain information is explicitly represented in a possibilistic ontology. That is, axioms with a null weight ($W(\varphi) = 0$) are not explicitly represented in the knowledge base. The weighted axiom $(\varphi, W(\varphi))$ means that the certainty degree of φ is at least equal to $W(\varphi)$. A possibilistic DL ontology Σ will also be represented by a pair $\Sigma = (T, A)$ where elements in both T and A may be uncertain. Note that if we consider all $W(\varphi_i) = 1$, then we find a classical DL ontology $\Sigma^* = \{\varphi_i : (\varphi_i, W(\varphi_i)) \in \Sigma\}$. We say that Σ is consistent if the classical ontology obtained from Σ by ignoring the weights associated with axioms is consistent, and inconsistent otherwise. Notice that the weights $W(\cdot)$ for axioms must be provided by the knowledge engineer that designs the knowledge base.

Introducing Possibilistic Description Logic Ontologies with Disjunctive Assertions

We now introduce the concept of possibilistic description logic ontology with disjunctive assertions. These new kind of ontologies are the possibilistic DL ontologies introduced above in Section 2 but with axioms formed in a particular way allowing to translate them into equivalent logic program rules and with assertional statements that

encode uncertainty about the membership of an individual to a disjunction of concepts. Gómez et al. [16, 17, 18] exploited the Description Logic Programming approach for translating DL ontologies into logic programming rules to reason on inconsistent DL ontologies in DeLP and PDeLP. Although these previous works handle a big number of cases, they are not able to handle the kinds of ontologies we introduce in this work and for this we present a new reasoning framework in Section 3.

The form of DL ontologies belonging to the DLP fragment comply with certain restrictions that make them to be successfully expressed in a logic programming setting [20]. Our aim is to translate inclusion axioms $C \sqsubseteq D$ into logic programming rules $d(X) \leftarrow c(X)$. $\mathcal{L}_b$-classes C can appear in the body of logic programming rules, i.e. in the left-hand side of DL inclusion axioms $C \sqsubseteq D$; $\mathcal{L}_h$-classes D can appear in the head of logic programming rules (e.g. they cannot be existentially quantified), so they appear in the right-hand side of DL inclusion axioms $C \sqsubseteq D$; finally, as DL equality axioms $C \equiv D$ are more restrictive, $\mathcal{L}_{hb}$-classes C and D can appear in the head and the body of rules.

Definition 1 *Let $C, C_1, \ldots, C_n$ be $\mathcal{L}_b$-classes, D an $\mathcal{L}_h$-class, A, B $\mathcal{L}_{hb}$-classes, $P, P_1, \ldots, P_n, Q$ properties, a, b individuals. Let T be a set of pairs $(\varphi, W(\varphi))$ whose first component is an inclusion or an equality axiom φ in $\mathcal{L}_{DL}$ coupled with a second component $W(\varphi)$ which is a possibilistic degree between 0 and 1. The axioms φ are of the form $C \sqsubseteq D$, $A \equiv B$, $\top \sqsubseteq \forall P.D$, $\top \sqsubseteq \forall P^-.D$, $P \sqsubseteq Q$, $P \equiv Q$, $P \equiv Q^-$, or $P^+ \sqsubseteq P$. Let A be a set of concept and role assertions $(\varphi, W(\varphi))$ disjoint with T of the form $a : (C_1 \sqcup \ldots \sqcup C_n)$ or $\langle a, b\rangle : (P_1 \sqcup \ldots \sqcup P_n)$. A* suppositional description logic ontology Σ *is a pair (T, A). The set T is called the* possibilistic terminology *(or just possibilistic Tbox), and A the* suppositional possibilistic assertional box *(or just supossitional possibilistic Abox).*

Next we present an example that is based on [10]. We will use this as a running example to show how the approach proposed handles the instance checking reasoning task in the presence of disjunction in assertional boxes of possibly inconsistent ontologies.

Example 1 *Consider the ontology $\Sigma_1 = (T, A)$ that expresses that ambulances and school buses are vehicles, regular vehicles are not allowed to park unless they are ambulances or school buses; also it is known that* a *is an ambulance or a school bus, and also a vehicle:*

$$T = \left\{ \begin{array}{l} (\mathsf{Ambulance} \sqcup \mathsf{SchoolBus} \sqsubseteq \mathsf{Vehicle}, 1), \\ (\mathsf{Vehicle} \sqsubseteq \neg\mathsf{Parking}, 0.6), \\ (\mathsf{Ambulance} \sqcup \mathsf{SchoolBus} \sqsubseteq \mathsf{Parking}, 0.9) \end{array} \right\}$$

$$A = \left\{ \begin{array}{l} (\mathsf{a} : (\mathsf{Ambulance} \sqcup \mathsf{SchoolBus}), 1), \\ (\mathsf{a} : \mathsf{Vehicle}, 1) \end{array} \right\}$$

We now explain intuitively why Σ_1 is inconsistent considering Σ_1^*. Suppose that a is an ambulance, then a is a member of Parking. But as ambulances are vehicles, then a is also a vehicle. Therefore a is also a member of $\neg$Parking. Then a is a member of Parking $\sqcap$ $\neg$Parking, i.e. a is a member of $\bot$. Absurd. The same reasoning applies when we suppose that a is a school bus. Next we will present a framework that will allow to deal with the situations introduced here in an elegant, natural, simple way.

3. REASONING WITH ONTOLOGIES HAVING DISJUNCTIVE ASSERTIONS IN SUPPOSITIONAL ARGUMENTATION SYSTEMS

Here we deal with the problem of reasoning with possibilistic description logic ontologies with disjunctive assertions (PoDLoDA ontologies for short). For this, we first introduce suppositional possibilistic defeasible logic programming (SPDeLP) that is a reasoning framework that we created; it allows to deal with uncertain information codified as suppositions and statements with numerical degrees of certainty. This framework is adaptation of Bodanza's suppositional argument system [10]. Second, we show PoDLoDA ontologies are expressed as SPDeLP programs and how reasoning task in ontologies are interpreted in this framework. Finally, we discuss differences and similarities of our interpretation of suppositional argumentation with the original framework proposed by Bodanza.

SPDeLP: Suppositional Possibilistic Defeasible Logic Programming

Suppositional argumention systems (SAS) introduced by Bodanza [10] provide a foundation for dealing intuitively with disjunctive information in a defeasible reasoning framework. Bodanza's view is that suppositional reasoning is present in defeasible arguments involving disjunctions, just as reasoning by cases appears in classical logic. Disjunctive information can express different plausible alternatives, and SAS study in what extent an argument assuming such possible alternatives can be considered relevant on the basis of its explicative power in comparison with other explanations. In this work we will present a adaptation of Bodanza's SAS to put emphasis in its implementability, thus providing a simpler presentation. We will adapt the language of Possibilistic Defeasible Logic Programming (PDeLP) [1] for allowing to represent facts having disjunctive literals.

A *suppositional possibilistic defeasible logic (SPDeLP) program* $\mathcal{P}$ is a set of rules $(P \leftarrow Q_1, \ldots, Q_n, \alpha)$ where $0 < \alpha \leq 1$ is a possibilistic degree. When $n = 0$, we will note them simply as (P, α). Facts have the form $(P_1 \text{ OR } \ldots \text{ OR } P_k, \alpha)$, where the P_i are either unary or binary predicates, meaning that at least one $P_i, i = 1, \ldots, k$ holds with degree α (when $k = 1$ we have traditional PDeLP facts). $P, Q_1, \ldots, Q_n$ are called literals and can be positive or negative atoms (i.e. classically negated with $\sim$). A set of rules is contradictory iff it allows to derive a pair of complementary literals L and $\sim L$ with some degree α_1 and α_2, resp. The tentative conclusions of the system are called, as usual, *arguments* and are built by the rules of derivation presented below.

Definition 2 *Let $\mathcal{P}$ be a SPDeLP program and α be a real number such that $0 < \alpha \leq 1$. An* argument *is a structure $\langle \mathcal{D}, \mathcal{S}, H, \alpha\rangle$, where $\mathcal{D}$ is a finite set of ground instances of rules in $\mathcal{P}$ (called the argument's* defeasible support*),*

(1) $(vehicle(X) \leftarrow ambulance(X), 1)$
(2) $(vehicle(X) \leftarrow schoolbus(X), 1)$
(3) $(\sim parking(X) \leftarrow vehicle(X), 0.6)$
(4) $(parking(X) \leftarrow ambulance(X), 0.9)$
(5) $(parking(X) \leftarrow schoolbus(X), 0.9)$
(6) $(ambulance(a)$ OR $schoolbus(a), 1)$
(7) $(vehicle(a), 1)$

Figure 1: Program $\mathcal{P}_1$ from Example 2

$\mathcal{S}$ is a set of ground literals called suppositions, *and H is a ground literal called the argument* conclusion *such that $\mathcal{D} \cup \mathcal{S}$ allows to derive H with strength α derived minimally according to the following deductive rules:*

- ***Fact (Fact):*** *If $(H, \alpha) \in \mathcal{P}$, then it holds that $\langle \emptyset, \emptyset, H, \alpha \rangle$ is an argument.*
- ***Supposition (Sup):*** *If $(H_1$ OR $\ldots$ OR $H_n, \alpha) \in \mathcal{P}$, then it holds that $\langle \emptyset, \{H_i\}, H_i, \alpha \rangle$ are arguments for $i = 1, \ldots, n$.*
- ***Generalized Modus Ponens (GMP):*** *If $\langle \mathcal{D}_1, \mathcal{S}_1, B_1, \alpha_1 \rangle$, $\ldots$, $\langle \mathcal{D}_n, \mathcal{S}_n, B_n, \alpha_n \rangle$, $(H \leftarrow B_1, \ldots, B_n, \alpha) \in \mathcal{P}$ and $(\bigcup_{i=1}^{n} \mathcal{D}_i) \cup (\bigcup_{i=1}^{n} \mathcal{S}_i) \cup \{H \leftarrow B_1, \ldots, B_n\}$ is not contradictory, then $\langle \bigcup_{i=1}^{n} \mathcal{D}_i \cup \{H \leftarrow B_1, \ldots, B_n\}, \bigcup_{i=1}^{n} \mathcal{S}_i, H, min(\alpha_1, \ldots, \alpha_n, \alpha) \rangle$ is an argument.*
- ***Constructive Dilemma (CD):*** *If $\langle \mathcal{D}_i, \{H_i\} \cup \mathcal{S}_i, B, \alpha_i \rangle$ for $i = 1, \ldots, n$ and $(H_1$ OR $\ldots$ OR $H_n, \alpha) \in \mathcal{P}$, and $(\bigcup_{i=1}^{n} \mathcal{D}_i) \cup (\bigcup_{i=1}^{n} \mathcal{S}_i)$ is not contradictory, then $\langle \bigcup_{i=1}^{n} \mathcal{D}_i, \bigcup_{i=1}^{n} \mathcal{S}_i, B, min(\alpha_1, \ldots, \alpha_n, \alpha) \rangle$ is an argument.*

Notice that we propose the CD rule based on the intuition that propositional logics allows to infer r from $p \vee q$, $p \rightarrow r$ and $q \rightarrow r$. Besides, we will say that an argument $\langle \mathcal{D}, \mathcal{S}, H, P \rangle$ is *self-defeating* iff $\mathcal{D} \cup \mathcal{S}$ allows to derive a pair of contradictory literals. Notice that, by construction, there are no self-defeating arguments in SPDeLP.

We refine the usual notion of disagreement: a set of arguments S is in disagreement if there are at least two arguments for some L and $\sim L$ in S that *have the same set of suppositions*. If S is not in disagreement, it is said to be in agreement. The conclusions of the system are obtained in the same way as in traditional PDeLP—through a dialectical process that considers all the arguments for a query, then all of its defeaters and defeaters for those defeaters and so on. Given two (contradicting) arguments, if the attacking argument is strictly preferred over the attacked one (i.e. its weight is greater), then it is called a *defeater*. Given a SPDeLP program $\mathcal{P}$ and a query H, the final answer to H w.r.t. $\mathcal{P}$ is based on such dialectical analysis. The answer to a query can be: *Yes* (when there exists a warranted argument $\langle \mathcal{D}, \mathcal{S}, H, \alpha \rangle$), *No* (when there exists a warranted argument $\langle \mathcal{D}, \mathcal{S}, \sim H, \alpha \rangle$), *Undecided* (when neither $\langle \mathcal{D}, \mathcal{S}, H, \alpha \rangle$ nor $\langle \mathcal{D}, \mathcal{S}, \sim H, \alpha \rangle$ are warranted), or *Unknown* (when H does not belong to $\mathcal{P}$).

We now introduce how the ontology presented in Example 1 is treated in SPDeLP. In Section 3, we will discuss how the translation from DL to SPDeLP is achieved.

Example 2 *(Continues Example 1) The ontology Σ_1 is interpreted as the program $\mathcal{P}_1$ in Fig. 1. Some of the arguments we can build from this program are:*

(8) $\mathcal{A}_1 = \langle \emptyset, \emptyset, vehicle(a), 1 \rangle$ *(by Fact from (7))*

(9) $\mathcal{A}_2 = \langle \emptyset, \{ambulance(a)\}, ambulance(a), 1 \rangle$ *(by Sup from (6))*

(10) $\mathcal{A}_3 = \langle \emptyset, \{schoolbus(a)\}, schoolbus(a), 1 \rangle$ *(by Sup from (6))*

(11) $\mathcal{A}_4 = \langle \{\sim parking(a) \leftarrow vehicle(a)\}, \emptyset, \sim parking(a), 0.6 \rangle$ *(by GMP from (3) and (8))*

(12) $\mathcal{A}_5 = \langle \{(parking(a) \leftarrow ambulance(a)\}, \{ambulance(a)\}, parking(a), 0.9 \rangle$ *(by GMP from (4) and (9))*

(13) $\mathcal{A}_6 = \langle \{parking(a) \leftarrow schoolbus(a)\}, \{schoolbus(a)\}, parking(a), 0.9 \rangle$ *(by GMP from (5) and (10))*

(14) $\mathcal{A}_7 = \langle \{parking(a) \leftarrow ambulance(a), parking(a) \leftarrow schoolbus(a)\}, \emptyset, parking(a), 0.9 \rangle$ *(by CD from (6), (12) and (13)).*

We see that $\mathcal{A}_5$, $\mathcal{A}_6$ and $\mathcal{A}_7$ are in agreement and disagree with $\mathcal{A}_4$. Argument $\mathcal{A}_7$ defeats argument $\mathcal{A}_4$, therefore a can park because $\mathcal{A}_4$ is warranted. Notice that $\mathcal{A}_5$ and $\mathcal{A}_6$ cannot attack $\mathcal{A}_4$ because, even entailing opposing conclusions, the set of suppositions for each argument are different.

Interpreting Suppositional Possibilistic DL Ontologies in SPDeLP

For assigning semantics to a description logics ontology, we define a translation function $\mathcal{T}(\cdot)$ from DL to SPDeLP based on the work of [20] (for details, see [17] to study the case in which ontologies belonging to the DLP fragment of DL are translated into DeLP). First, axioms are considered to be in negation-normal form, meaning that negations are pushed inward class expressions. Informally, an axiom of the form $\mathsf{C} \sqsubseteq \mathsf{D}$ will be translated as $d(X) \leftarrow c(X)$. Abox assertions of the form $\mathsf{a} : (\mathsf{C}_1 \sqcup \ldots \sqcup \mathsf{C}_n)$ are translated as facts $c_1(a)$ OR $\ldots$ OR $c_n(a)$ and $\langle \mathsf{a}, \mathsf{b} \rangle : (\mathsf{r}_1 \sqcup \ldots \sqcup \mathsf{r}_n)$ as $r_1(a, b)$ OR $\ldots$ OR $r_n(a, b)$. Moreover, a formula of the form $\exists \mathsf{r}.\mathsf{C} \sqsubseteq \mathsf{D}$ is translated as $d(X) \leftarrow r(X, Y), c(Y)$, and one of the form $C \sqcup D \sqsubseteq E$ as two axioms $C \sqsubseteq E$ and $D \sqsubseteq E$. See Fig. 2 for a formal account of the translation function. The interpretation of Σ is a SPDeLP program $\mathcal{P} = \mathcal{T}(T) \cup \mathcal{T}(A)$.

We redefine instance checking to handle possible inconsistencies while retaining classical DL functionality: If C is a class, a an individual and α a real number between 0 and 1, then (i) a is a potential member of C iff there exists an argument $\langle \mathcal{A}, \emptyset, C(a), \alpha \rangle$ w.r.t. $\mathcal{P}$, and, (ii) a is a justified member of C iff there exists a warranted argument $\langle \mathcal{A}, \emptyset, C(a), \alpha \rangle$ w.r.t. $\mathcal{P}$.

Example 3 *(Continues Example 2) Consider again Σ_1. The individual* a *is a justified member of the concept* Parking.

$$
\begin{array}{rcl}
\mathcal{T}(\{C \sqsubseteq D\}) & =_{df} & \{\ T_h(D,X) \leftarrow T_b(C,X)\ \}, \text{if } C \text{ is an } \mathcal{L}_b\text{-class and } D \text{ an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(\{C \equiv D\}) & =_{df} & \mathcal{T}(\{C \sqsubseteq D\}) \cup \mathcal{T}(\{D \sqsubseteq C\}), \text{if } C \text{ and } D \text{ are } \mathcal{L}_{hb}\text{-classes} \\
\mathcal{T}(\{\top \sqsubseteq \forall P.D\}) & =_{df} & \{\ T_h(D,Y) \leftarrow P(X,Y)\ \}, \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(\{\top \sqsubseteq \forall P^{-}.D\}) & =_{df} & \{\ T_h(D,X) \leftarrow P(X,Y)\ \}, \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(\{a : (D_1 \sqcup \ldots \sqcup D_n)\}) & =_{df} & \{\ T_h(D_1,a)\ \mathsf{OR}\ \ldots\ \mathsf{OR}\ T_h(D_n,a)\ \}, \text{if } D_i \text{ are } \mathcal{L}_h\text{-classes} \\
\mathcal{T}(\{\langle a,b\rangle : (P_1 \sqcup \ldots \sqcup P_n)\}) & =_{df} & \{\ P_1(a,b)\ \mathsf{OR}\ \ldots\ \mathsf{OR}\ P_n(a,b)\ \} \\
\mathcal{T}(\{P \sqsubseteq Q\}) & =_{df} & \{\ Q(X,Y) \leftarrow P(X,Y)\ \} \\
\mathcal{T}(\{P \equiv Q\}) & =_{df} & \left\{ \begin{array}{l} Q(X,Y) \leftarrow P(X,Y) \\ P(X,Y) \leftarrow Q(X,Y) \end{array} \right\} \\
\mathcal{T}(\{P \equiv Q^{-}\}) & =_{df} & \left\{ \begin{array}{l} Q(X,Y) \leftarrow P(Y,X) \\ P(Y,X) \leftarrow Q(X,Y) \end{array} \right\} \\
\mathcal{T}(\{P^{+} \sqsubseteq P\}) & =_{df} & \{\ P(X,Z) \leftarrow P(X,Y) \wedge P(Y,Z)\ \} \\
\mathcal{T}(\{s_1,\ldots,s_n\}) & =_{df} & \bigcup_{i=1}^{n} \mathcal{T}(\{s_i\}), \text{if } n > 1 \\
\textbf{where:} & & \\
T_h(A,X) & =_{df} & A(X) \\
T_h((C \sqcap D),X) & =_{df} & T_h(C,X) \wedge T_h(D,X) \\
T_h((\forall R.C),X) & =_{df} & T_h(C,Y) \leftarrow R(X,Y) \\
T_b(A,X) & =_{df} & A(X) \\
T_b((C \sqcap D),X) & =_{df} & T_b(C,X) \wedge T_b(D,X) \\
T_b((C \sqcup D),X) & =_{df} & T_b(C,X) \vee T_b(D,X) \\
T_b((\exists R.C),X) & =_{df} & R(X,Y) \wedge T_b(C,Y)
\end{array}
$$

Figure 2: Mapping $\mathcal{T}$ from DL ontologies with assertions to SPDeLP rules

Instance Checking with Contrapositive Reasoning

The approach is presented so far does not allow to perform contrapositive reasoning (i.e. we cannot deduce that a is an instance of $\neg C$ from $C \sqsubseteq D$ and $a : \neg D$). The simplest solution to this problem is to consider adding transposes of rules for so-called strict rules (i.e. rules with weight equal to 1), a resource that will allow us to perform reasoning by contrapositive reasoning while retaining the implementability of the system. This feature is based on approaches such as Vreeswijk's system ASPIC (see `http://aspic.cossac.org/`) and Alsinet et al. [1].

Definition 3 *Let $r = (H \leftarrow B_1, B_2, \ldots, B_n, \alpha)$ be a rule. The* set of transposes of r, *denoted as* $\mathsf{Trans}(r)$, *is*

$$
\mathsf{Trans}(r) \quad = \quad \left\{ \begin{array}{l} (H \leftarrow B_1, B_2, \ldots, B_n, \alpha), \\ (\sim B_1 \leftarrow \sim H, B_2, \ldots, B_n, \alpha), \\ (\sim B_2 \leftarrow B_1, \sim H, \ldots, B_n, \alpha), \\ \ldots \\ (\sim B_n \leftarrow B_1, B_2, \ldots, \sim H, \alpha) \end{array} \right\}
$$

Example 4 *Consider again the axiom* $(\mathsf{Ambulance} \sqcup \mathsf{SchoolBus} \sqsubseteq \mathsf{Vehicle}, 1)$ *from Example 1. According to the translation function $\mathcal{T}(\cdot)$ in Figure 2, by Lloyd-Topor transformations, we obtain rules (1) and (2) from the axioms* $(\mathsf{Ambulance} \sqsubseteq \mathsf{Vehicle}, 1)$ *and* $(\mathsf{SchoolBus} \sqsubseteq \mathsf{Vehicle}, 1)$ *(see program $\mathcal{P}_1$ in Figure 1). Yet if we had $(b : \neg\mathsf{Vehicle}, 0.9)$ as an additional assertional expression in A, we would not be able to deduce that b is not a member of* Ambulance *solely from rules (1) and (2). Instead, by considering their transposes, viz. $(\sim ambulance(X) \leftarrow\sim vehicle(X), 1)$ and $(\sim schoolbus(X) \leftarrow\sim vehicle(X), 1)$, by GMP we can infer $(\sim ambulance(b), 0.9)$, thus concluding that b is not an ambulance with strength 0.9.*

4. RELATED WORK

Comparison with Bodanza's SAS: Here we briefly discuss how our proposal for suppositional argumentation differs from the one presented by Bodanza in [10]. Bodanza includes a *deduction* rule that is too powerful to be implemented because it relies on deduction in first-order classical logic (see [10, p. 27]). We instead chose to include a simpler rule such as CD. We also chose to drop the *conditionalization* rule of Bodanza; this makes our system less expressive in the sense that we cannot discharge suppositions (except by using CD). However, adding transposes of strict rules allows to perform contrapositive reasoning without recurring to first-order logic rules. We use weights attached to rules for ultimately comparing the strength of arguments but Bodanza chooses to use generalized specificity as a comparison criterion. It can be argued that the inclusion of numerical weights in our system obscures the knowledge representation but we think that this feature allows the knowledge engineer to have more control and simplifies the implementation of the reasoning system.

Disjunction in argumentation: Reasoning with disjunctive knowledge bases is an interesting research topic that has been addressed in the past by earlier nonmonotonic reasoning systems such as default logic [15, 28]. Wang and Chen [29, 30] study the relationship existing between abduction-based argumentation and disjunctive logic programming. In their proposal, disjunctions of negative literals are regarded as possible assumptions and are used to represent incomplete information. They define three semantics corresponding to credulous, moderate and skeptical reasoning, resp. They claim to be the first argumentation-based abductive approach applied to disjunctive logic programs. Wang and Chen consider full disjunctive logic programming rules (i.e. rules with disjunctions in the head and non-empty bodies); our approach, however, is preliminary in that sense because we only consider facts with disjunctions (i.e. rules with disjunctions in the head and empty bodies). Moreover, considering the

three main kinds (in a philosophical sense) of reasoning (viz., deductive, inductive and abductive) our approach is deductive and Wang's is abductive.

Bochman [9] introduces an extension of an abstract argumentation framework that provides a direct representation of global conflicts between sets of arguments. The extension, called collective argumentation, is suitable for representing semantics of disjunctive logic programs. Collective argumentation theories possess a four-valued semantics as our system does. Two special kinds of collective argumentation, positive and negative argumentation, are considered in which the opponents can share their arguments. Negative argumentation turns out to be especially appropriate for analyzing stable sets of arguments. Positive argumentation generalizes certain alternative semantics for logic programs. In Bochman's examples, only disjunctive facts are considered and default negation is used instead of classical negation. Bochman's attack relationship allows for collective attacks among arguments; our approach in turn only allows for pairwise attack between arguments.

Nieves et al. [26] present a possibilistic disjunctive logic programming approach where its semantics is characterized by a fixed-point operator. To manage inconsistency, they define a preference criterion to decide between inconsistent possibilistic models and use cuts to restore consistency. The proposal of Nieves et al. is more expressive than ours because full disjunctive rules are allowed and not only disjunctive facts. However, we accept inconsistency and deal with it with an argumentative process instead of repairing a disjunctive program.

Inconsistency handling in weighted ontologies: Beferhat and Bouraoui [5] affirm that Hollunder [21] was the first in proposing combining possibilistic logics and ontologies and then Dubois et al. [13] discussed again the idea. Gómez et al. [16] presented a reasoning framework for dealing with possibly inconsistent ontologies represented in the description logic programming (DLP) fragment of DL based on possibilistic defeasible logic programming known as weighted ontologies. They handle uncertainty by adding possibilistic weights to rules like we do.

Lukasiewicz [23] presents the expressive probabilistic description logics P-$\mathcal{SHIF}(\mathbf{D})$ and P-$\mathcal{SHOIN}(\mathbf{D})$, which are probabilistic extensions of the $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$ description logics, resp. They are semantically based on the notion of probabilistic lexicographic entailment from probabilistic default reasoning, which interprets terminological and assertional probabilistic knowledge about random and concrete instances resp., which is semantically interpreted as in Lehmann's lexicographic entailment in default reasoning from conditional knowledge bases. Possibilistic approaches to reasoning associate each piece of knowledge with a certainty degree. Lukasiewicz, instead, proposes conditional probability statements having a probability range.

Benferhat and Bouraoui [5, 6] present a possibilistic version of DL-Lite, called π-DL-Lite, suitable for handling inconsistency. We deal with disjunction which is not allowed DL-Lite, and, despite the simplicity of the examples presented here, our framework is targeted to the DLP fragment of DL which allows to handle richer knowledge representations. Nonetheless, DL-Lite allows the equivalent of existential quantifiers in the head of rules, a feature that logic programming does not allow. In particular in [7], they only deal with inconsistency in the Abox and not in the Tbox. On the contrary, we assume a consistent Abox having disjunctive assertions and the inconsistency is given by considering the information in the Tbox along with the Abox.

5. CONCLUSIONS AND FUTURE WORK

This paper proposed a kind of DL ontologies that allow to represent disjunctive assertions of membership of individuals to concepts and roles where the ontology axioms have been assigned possibilistic degrees of certainty, and also proposed a variation of suppositional argumentation to handle the reasoning task of instance checking in such ontologies. Inconsistent ontologies are interpreted into a possibilistic suppositional argumentation system redefining instance checking via a warrant procedure in such system. For every disjunctive assertion, if the supposition of each disjunct allows the system to reach a certain conclusion, then that conclusion can be taken for granted. Because the ontology can be potentially inconsistent, the reasoning framework takes this possibility into account to compute defeaters and defeaters for these defeaters to perform a dialectical analysis to warrant conclusions. In our approach, the possibilistic degrees of certainty are used to compute the relative weight of each possible conclusion, which, in turn, are used to decide which arguments prevail. Much work remains to be done such us proposing larger case studies. With respect to knowledge representation aspects, we would like to adapt the system to have disjunctions in the head of rules. Also considering different argumentation semantics for the definition of the dialectical process appears to be a promising avenue of research.

Acknowledgments: This research is funded by Secretaría General de Ciencia y Técnica, Universidad Nacional del Sur, Argentina.

References

[1] Teresa Alsinet, Carlos Iván Chesñevar, and Lluis Godo. A level-based approach to computing warranted arguments in possibilistic defeasible logic programming. In Philippe Besnard, Sylvie Doutre, and Anthony Hunter, editors, *COMMA*, volume 172 of *Frontiers in Artificial Intelligence and Applications*, pages 1–12. IOS Press, 2008.

[2] Grigoris Antoniou and Antonis Bikakis. DR-Prolog: A System for Defeasible Reasoning with Rules and Ontologies on the Semantic Web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):233–245, 2007.

[3] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook – Theory, Implementation and Applications*. Cambridge University Press, 2003.

[4] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[5] Salem Benferhat and Zied Bouraoui. Possibilistic DL-Lite. In *Scalable Uncertainty Management*, pages 346–359. Springer, 2013.

[6] Salem Benferhat and Zied Bouraoui. Min-based possibilistic dl-lite. *Journal of Logic and Computation*, 2015.

[7] Salem Benferhat, Zied Bouraoui, Sylvain Lagrue, and Julien Rossit. Merging Inconmensurable Possibilistic DL-Lite Assertional Bases. In Odile Papini, Salem Benferhat, Laurent Garcia, and Marie-Laure Mugnier, editors, *Proceedings of the IJCAI Workshop 13 Ontologies and Logic Programming for Query Answering*, pages 90–95, 2015.

[8] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.

[9] Alexander Bochman. Collective Argumentation and Disjunctive Logic Programming. *Journal of Logic and Computation*, 13(3):405–428, 2003.

[10] Gustavo Bodanza. Disjunctions and Specificity in Suppositional Defeasible Argumentation. *Logic Journal of the Interest Group in Pure and Applied Logics*, 10(1):23–49, 2002.

[11] Martin Caminada and Leila Amgoud. On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171:286–310, 2007.

[12] Carlos Iván Chesñevar, Ana Maguitman, and Ronald Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.

[13] D. Dubois, J. Mengin, and H. Prade. Possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web. Capturing Intelligence*, volume 1, pages 101–113. Elsevier, 2006.

[14] A. García and G. Simari. Defeasible Logic Programming an Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[15] Michael Gelfond, Vladimir Lifschitz, Halina Przymusińska, and Mirolaw Truszczyński. Disjunctive Defaults. In *Proc. 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 230–237, 1991.

[16] Sergio A. Gómez, Carlos I Chesñevar, and Guillermo R. Simari. Using Possibilistic Defeasible Logic Programming for Reasoning with Inconsistent Ontologies. In Armando Di Giusti and Javier Diaz, editors, *Computer Science & Technology Series. XVII Argentine Congress of Computer Science Selected Papers*, pages 19–29, 2012.

[17] Sergio Alejandro Gómez, Carlos Iván Chesñevar, and Guillermo Ricardo Simari. Reasoning with Inconsistent Ontologies Through Argumentation. *Applied Artificial Intelligence*, 1(24):102–148, 2010.

[18] Sergio Alejandro Gómez, Carlos Iván Chesñevar, and Guillermo Ricardo Simari. ONTOarg: A Decision Support Framework for Ontology Integration based on Argumentation. *Expert Systems with Applications*, 40:1858–1870, 2013.

[19] Sergio Alejandro Gómez and Guillermo Ricardo Simari. Merging of ontologies using belief revision and defeasible logic programming. *Inteligencia Artificial*, 16(52):16–28, 2013.

[20] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description Logic Programs: Combining Logic Programs with Description Logics. *WWW2003, May 20-24, Budapest, Hungary*, 2003.

[21] B. Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. *International Journal of Approximate Reasoning*, 12(2):85–109, 1995.

[22] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with Inconsistent Ontologies. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 454–459, Edinburgh, Scotland, August 2005.

[23] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172:852–883, 2008.

[24] Martín O. Moguillansky and Marcelo A. Falappa. A non-monotonic Description Logics model for merging terminologies. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 11(35):77–88, 2007.

[25] Martín O. Moguillansky, Nicolás D. Rotstein, and Marcelo A. Falappa. Generalized Abstract Argumentation: A First-order Machinery towards Ontology Debugging. *Inteligencia Artificial*, 46:17–33, 2010.

[26] Juan Carlos Nieves, Mauricio Osorio, and Ulises Cortés. Semantics for Possibilistic Disjunctive Programs. *Theory and Practice of Logic Programming (TPLP)*, 13(1):33–70, 2013.

[27] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.

[28] Chiaki Sakama and Katsumi Inoue. Relating disjunctive logic programs to default theories. In *Proceedings of the 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning (LPNMR'93)*, pages 266–282. MIT Press, 1993.

[29] Kewen Wang. Argumentation-based abduction in disjunctive logic programming, 2000.

[30] Kewen Wang and Lizhu Zhou. Comparisons and computation of well-founded semantics for disjunctive logic programs. *ACM Trans. Comput. Logic*, 6(2):295–327, April 2005.