A Preliminary Framework for 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
Email: sag@cs.uns.edu.ar

Abstract. We present a preliminary framework for reasoning with *possibilistic description logics ontologies with disjunctive assertions* (PDLDA ontologies for short). PDLDA ontologies are composed of a terminology as well as an assertional box that allows to declare three kinds of assertional statements: an individual is a member of one 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*. Each axiom in the ontologies has a certainty degree as is usual in possibilistic logics. For reasoning with PDLDA ontologies, we interpret them in terms of a adaptation of Bodanza's Suppositional Argumentation System. Our framework allows to reason with modus ponens and constructive dilemmas. We use it for determining the membership of individuals to concepts when there is doubt to exactly which one of the concepts in the union the individual belongs. 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: Supppositional argumentation, ontology reasoning, inconsistency handling, Description Logics.

1 Introduction

Reasoning with Description Logics ontologies [1] is an important topic for the implementation of the Semantic Web [2], where data resources are described in terms of ontologies expressed in the Web Ontology Language (OWL-DL). OWL-DL underlying semantics are expressed in Description Logics. Traditionally, an ontology is a set of axioms that defines 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).

Incoherence and inconsistency are two anomalies that may affect an ontology. An incoherent ontology has definitions that render some concepts empty.

Inconsistent ontologies imply that an individual is a member of both a concept and its complement. Because DL ontologies are based on decidable fragments of first-order logic, inconsistency makes anything derivable from such an ontology.

Inconsistent ontologies pose a problem for this because they have to be debugged by the knowledge engineer prior to deployment of applications. Many times it is not possible to do this because the domain being modeled is intrinsically contradictory or the programmer does not have the authority to edit the ontology's contents. Many approaches have been proposed for dealing with this situation is solved in two main ways. The first kind of approaches consists of eliminating part of the ontology to make it consistent again (e.g. approaches based on Belief Revision theory [3, 4] or on paraconsistent logics [5]). The second kind consists of accepting inconsistency and using some kind of non-standard reasoning mechanism to get some meaningful answers from an inconsistent knowledge base (e.g. argumentation [6–9]). We will rely on the latter approach in this paper.

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 [10–12]. Given a claim, this process usually takes into account defeaters (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) [13] is a rule-based implementation of defeasible argumentation based on the Prolog programming language. Possibilistic DeLP (or just PDeLP) [14] is another argumentative approach to 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 priority, thus imposing a rule priority principle for comparing arguments based on the idea that an argument is as credible as the least credible of the rules supporting it.

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, in this paper, 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). SPDeLP is an adaptation of Bodanza's Suppositional Argumentation System [15] for making it suitable for its 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"*. 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 *PDLDA ontologies*. We then interpret PDLDA ontologies as SPDeLP programs in order to answer queries of membership of individual to classes in the presence of inconsistency. We think that this

approach could extend the range of applications in which ontologies are used, such as improving the development of applications that need dealing with uncertain knowledge to perform decision-making. We present a running example that shows how SPDeLP allows to reason on a given PDLDA ontology.

The rest of the article is structured as follows. In Sect. 2, we present the fundamentals of PDLDA ontologies. In Sect. 3, we describe the argumentation approach to reasoning with SPDeLP and how it is used to perform instance checking in PDLDA ontologies. Finally, Sect. 4 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. First we briefly recall reasoning in description logics, second the variation proposed for possibilistic description logics, to finally present ontologies with disjunctive assertions.

2.1 A Brief Recall of Description Logics

Description Logics (DL) are a well-known family of knowledge representation formalisms [1]. 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 *Abox reasoning* tasks are defined in DL, 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 approaches [16] take advantage of this to interpret such axioms as the Prolog rules "d(X) :- c(X)." and "c(a).", resp. We will apply this later on 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 [7] for details).

2.2 Fundamentals of Possibilistic Description Logics

We now recall the fundamentals of possibilistic description logic ontologies. Our presentation is based on [17] and [18]. 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.

2.3 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 Sect. 2.2 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. [7, 8, 17] exploited the Description Logic Programming approach for translating DL ontologies into logic programming rules to reason on inconsistent DL ontologies in DeLP and PDeLP. These previous works, although they handle a big number of cases, are not able to handle the kind of ontologies we are going to propose here. Handling these kind of ontologies requires a new reasoning framework that we will introduce in Sect. 3.

To understand how the translation from DL to logic programming works, we recall some concepts presented in [7]. In the presentation, we will refer to $\mathcal{L}_b$-classes, $\mathcal{L}_h$-classes and $\mathcal{L}_{hb}$-classes. Remember that our aim is to translate inclusion axioms $C \sqsubseteq D$ into logic programming rules $d(X) \leftarrow c(X)$. Because

of the restrictive form of logic programming rules, $\mathcal{L}_b$-classes C are classes that can appear in the body of logic programming rules, that is 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 are such that they have to be able to appear in the head and the body of rules.

Definition 1. *Let $C, C_1, \ldots, C_n$ be an $\mathcal{L}_b$-class, 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).*

We will present now a paradigmatic example that is based on [15]. We will use this as a running example to show how the approach proposed in this work 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 except ambulances and school buses; also it is known that a is an ambulance or an school bus, and also a vehicle:

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

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

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

3 Reasoning with Ontologies with Disjunctive Assertions in Suppositional Argumentation Systems

In this section, we deal with the problem of reasoning with possibilistic description logic ontologies with disjunctive assertions (PDLDA 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 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 [15]. Second, we show PDLDA 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.

3.1 SPDeLP: Suppositional Possibilistic Defeasible Logic Programming

Suppositional argumention systems (SAS) introduced by Bodanza [15] 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) [14] 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 (that is 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*), $\mathcal{S}$ 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 $(H_1 \text{ OR } \ldots \text{ OR } H_n, \alpha) \in \mathcal{P}$, then it holds that $\langle \emptyset, \{H_i\}, H_i, \alpha \rangle$ are arguments for $i = 1, \ldots, n$.*

- ***Generalized Modus Ponens (GMP):*** *If* $\langle \mathcal{D}_1, \mathcal{S}_1, B_1, \alpha_1\rangle$, ..., $\langle \mathcal{D}_n, \mathcal{S}_n, B_n, \alpha_n\rangle$, $(H \leftarrow B_1, \ldots, B_n, \alpha) \in \mathcal{P}$ *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* $(H_1 \text{ OR } \ldots \text{ OR } H_n, \alpha) \in \mathcal{P}$, *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 the CD rule is proposed based on the intuition that in propositional logics r can be inferred from $p \vee q$, $p \rightarrow r$ and $q \rightarrow r$. Also, 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.

Property 1. 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 (that is 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 Ex. 1 is treated in SPDeLP. In Sect. 3.2, we will discuss how the translation from DL to SPDeLP is achieved.

(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) $(vehicle(a), 1)$

Fig. 1. Program $\mathcal{P}_1$ from Example 2

Example 2. (Continues Ex. 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.

3.2 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 [16] (for details, see [7] to study the case in which DL ontologies 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 $\mathcal{P} = \mathcal{T}(T) \cup \mathcal{T}(A)$.

Instance checking is redefined 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 Ex. 2) Consider again Σ_1. The individual a is a justified member of the concept $\mathsf{Parking}$.

3.3 Discussion

Here we briefly discuss how our proposal for suppositional argumentation differs from the one presented by Bodanza in [15]. Bodanza includes a *deduction* rule that is too powerful to be implemented because it relies in deduction in first-order classical logic (see [15, 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) and our system does not allow reasoning by contrapositive reasoning. Bodanza's approach does not use weights for rules but instead uses generalized specificity to compare arguments. Our inclusion of numerical weights obscures the knowledge representation but gives more control to the knowledge engineer and simplifies the implementation of the reasoning system.

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

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

4 Conclusions and Future Work

This paper proposed a kind of description logic 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. The idea is to translate the ontologies into a possibilistic suppositional argumentation system in order to perform the reasoning on the ontologies. For every disjunctive assertion, if supposing each disjunct allows the system to reach a conclusion, then that conclusion can be taken for granted. As 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 reasoning in order to warrant conclusions. 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. It also appears to be interesting to consider different argumentation semantics for the definition of the dialectical process.

Acknowledgments: This research is funded by Secretaría General de Ciencia y Técnica, Universidad Nacional del Sur, Argentina. The author would like to thank the anonymous reviewers for their comments.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook – Theory, Implementation and Applications. Cambridge University Press (2003)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
3. Moguillansky, M.O., Falappa, M.A.: A non-monotonic Description Logics model for merging terminologies. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial **11**(35) (2007) 77–88
4. Moguillansky, M.O., Rotstein, N.D., Falappa, M.A.: Generalized Abstract Argumentation: A First-order Machinery towards Ontology Debugging. Inteligencia Artificial **46** (2010) 17–33
5. Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with Inconsistent Ontologies. In Kaelbling, L.P., Saffiotti, A., eds.: Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, Scotland (August 2005) 454–459
6. Gómez, S.A., Simari, G.R.: Merging of ontologies using belief revision and defeasible logic programming. Inteligencia Artificial **16**(52) (2013) 16–28
7. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: Reasoning with Inconsistent Ontologies Through Argumentation. Applied Artificial Intelligence **1**(24) (2010) 102–148
8. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: ONTOarg: A Decision Support Framework for Ontology Integration based on Argumentation. Expert Systems with Applications **40** (2013) 1858–1870
9. Antoniou, G., Bikakis, A.: DR-Prolog: A System for Defeasible Reasoning with Rules and Ontologies on the Semantic Web. IEEE Transactions on Knowledge and Data Engineering **19**(2) (2007) 233–245
10. Chesñevar, C.I., Maguitman, A., Loui, R.: Logical Models of Argument. ACM Computing Surveys **32**(4) (December 2000) 337–383
11. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artificial Intelligence **171**(10-15) (2007) 619–641
12. Rahwan, I., Simari, G.R.: Argumentation in Artificial Intelligence. Springer (2009)
13. García, A., Simari, G.: Defeasible Logic Programming an Argumentative Approach. Theory and Practice of Logic Programming **4**(1) (2004) 95–138
14. Alsinet, T., Chesñevar, C.I., Godo, L.: A level-based approach to computing warranted arguments in possibilistic defeasible logic programming. In Besnard, P., Doutre, S., Hunter, A., eds.: COMMA. Volume 172 of Frontiers in Artificial Intelligence and Applications., IOS Press (2008) 1–12
15. Bodanza, G.: Disjunctions and Specificity in Suppositional Defeasible Argumentation. Logic Journal of the Interest Group in Pure and Applied Logics **10**(1) (2002) 23–49
16. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logics. WWW2003, May 20-24, Budapest, Hungary (2003)
17. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: Using Possibilistic Defeasible Logic Programming for Reasoning with Inconsistent Ontologies. In Giusti, A.D., Diaz, J., eds.: Computer Science & Technology Series. XVII Argentine Congress of Computer Science Selected Papers. (2012) 19–29
18. Benferhat, S., Bouraoui, Z., Lagrue, S., Rossit, J.: Merging Inconmensurable Possibilistic DL-Lite Assertional Bases. In Papini, O., Benferhat, S., Garcia, L., Mugnier, M.L., eds.: Proceedings of the IJCAI Workshop 13 Ontologies and Logic Programming for Query Answering. (2015) 90–95