A Nonmonotonic Rule System using Ontologies

G. Antoniou

Dept. of Computer Science, University of Bremen
ga@tzi.de

Abstract. The development of the Semantic Web proceeds in layers. Currently the most advanced layer that has reached maturity is the ontology layer, in the form of the DAML+OIL language which corresponds to a rich description logic. The next step will be the the realization of logical rule systems on top of the ontology layer.

Computationally simple nonmonotonic rule systems show promise to play an important role in electronic commerce on the Semantic Web. In this paper we show how nonmonotonic rule systems in the form of defeasible reasoning, can be built on top of description logics. The key idea is to use concept and role predicates in the antecedents of rules. We define a proof theory for this kind of reasoning, and derive some basic properties.

1 Introduction

The Semantic Web initiative [28] promises to improve dramatically the World Wide Web, and in doing so, to have significant impact on the way information is exchanged and business is conducted. The main idea is to use *machine processable* data and knowledge.

The development of the semantic web proceeds in layers, each layer being on top of lower layers. At present, the highest layer that has reached sufficient maturity is the *ontology layer* in the form of the DAML+OIL language [7]. DAML+OIL was designed to be sufficiently rich to be useful in applications, while being simple enough to allow for efficient reasoning support. In fact, it corresponds to an expressive *description logic*. Stated another way, the realization of the ontology layer could draw benefits from extensive previous work on description logics [6, 16, 15], both with regard to clear semantics and efficient reasoning support.

The next step in the development of the semantic web core will be the realization of the *logic* and *proof layers*. These layers will be built on top of ontology languages, and will offer enhanced representation and reasoning capabilities. A key ingredient of these layers will be *rules*, which look likely to become an action focus of the W3C. *Monotonic rule systems* are well known and widely in use. Seen as a subset of predicate logic (Horn logic), they are orthogonal to description logics: none is a proper subset of the other. The realization of a monotonic rule layer on top of the ontology layer can draw on work on hybrid reasoning, combining description logics with Horn logic, or at least Datalog [16, 9, 10].

But there exist other kinds of rule systems that are *nonmonotonic*. Such systems are important in practice because they can deal with inconsistencies in a declarative way, and because they model naturally phenomena like exceptions and priorities. In the past few years, such systems have attracted significant attention in the nonmonotonic reasoning community, e.g. courteous logic programs [12, 13] and defeasible logics [21, 1, 2]. Their use in various application domains has been advocated, including the modelling of regulations and business rules [20, 13, 3], modelling of contracts [13], legal reasoning [22] and agent negotiations [11]. In fact, defeasible reasoning (in the form of courteous logic programs [12, 13]) provides the foundation for IBM's Business Rules Markup Language and for current W3C activities on rules. Therefore defeasible reasoning is arguably one of the most successful subarea in nonmonotonic reasoning as far as applications and integration to mainstream IT is concerned.

One important advantage of such systems is their focus on implementability and their low computational complexity [18, 13]. Traditional nonmonotonic reasoning systems [19] have had a high computational complexity. This was justified by the desire to model interesting reasoning phenomena. Potential applications of such rich systems are in the areas of planning, cognitive robotics, scheduling etc. Still they fail to implement the original motivation of "jumping into conclusions", and they are too inefficient for time-critical systems.

Defeasible reasoning takes a different approach: simple and efficient. So seen, it makes sense to study the integration of description logics with defeasible reasoning, since both share a focus on efficiency. The integration of nonmonotonic rule systems with description logic based ontologies can serve two purposes: (1) Enhanced reasoning capabilities may be used to express richer ontological knowledge. For example, defeasible ontologies can be built, an idea that appears reasonable in the legal domain. (2) Rule- based systems define ontology-based applications using vocabulary defined in description logic. This idea is compatible with significant work on hybrid reasoning, e.g. [16].

In this paper we concentrate on the second approach, and we will study defeasible reasoning [21] running on top of description logics. Our task is more complex than the integration of Horn rules with description logics because of the following observation: both Horn logic and description logics are subsets of predicate logic. Therefore *semantically* there is no difficulty at all, and the focus of work on hybrid reasoning has been on efficient algorithms and limits of computability. However nonmonotonic rules are not a subset of predicate logic, and we need to define the semantics of such a combination, too. On the other hand, because nonmonotonic rules with variables are interpreted as schemas, their integration with description logics does not have difficulties encountered in other works on hybrid reasoning with regard to instantiation [9].

In this paper we define a proof theory, which can form the basis for efficient algorithms. Also we study some basic properties of the proof theory, as well as some variations of the key idea, following slightly different intuitions.

We should mention that the integration of terminological/description logics with other forms of nonmonotonic reasoning in hybrid systems has been studied before [4, 5, 23, 25]. Our work is novel in that defeasible logic is used as the default reasoning mechanism. Its choice seems natural since it has a low computational complexity which fits well with the potential Semantic Web applications, but is still sufficiently rich to be useful in business rules modelling and automated agent negotiations.

In work developed independently of ours, [14] also addresses combining non-monotonic rules with description logic ontology knowledge.

2 Nonmonotonic Reasoning and the Semantic Web: Why?

There has been some discussion in the Semantic Web community about what logic to use. Some people have claimed that there is no place for nonmonotonic reasoning. We believe that nonmonotonic rules will play an important role in the areas of (Semantic Web enabled) Electronic Commerce and Knowledge Management. In these areas situations arise naturally where the available information is incomplete; exactly the kinds of situations where nonmonotonic reasoning is useful. Let us briefly illustrate a scenario relevant to the Semantic Web.

Suppose that I have had so positive experiences with my Semantic Web personal agent that I trust it fully. I wake up in the morning thinking of my girlfriend who is in Greece, while I am in Germany, and decide to send her flowers. I ask the agent to do the job and leave home, being unavailable for further contact. The agent sets out to locate several relevant service companies and to compare their price, reputation, delivery policies etc.

Now suppose a company has a policy that it will grant a special 5% discount if the recipient happens to have birthday on that day. Further suppose the company is wise enough to represent its pricing policy in a declarative way. Now how could it represent the discounting rule? It cannot be sure to receive the information about birthday (unaware of the discounting possibility I failed to tell my personal agent; nor can I be contacted). This is a typical situation where reasoning must be made in the presence of incomplete information. Obviously the pricing policy needs something like the following:

R_1: If a birthday is provided and corresponds with the current date then give a 5% discount.
R_2: If the birthday is not provided then use the standard price.

The problem with R_2 is that it cannot be expressed properly in monotonic rule systems because it depends on the absence of information, not the falsity of a statement (it may be the birthday today, bit not known). If we use

R_2': If not birthday today then standard price.

instead, then R_2' cannot fire if we know for sure that it is not the recepient's birthday today. And if we use the rule

R_2'': Use the standard price.

then we will get an inconsisteny between R_1 and R_2'' in case it is the birthday. The solution using nonmonotonic rules is simple:

R_1: If a birthday is provided and corresponds with the current date then give a 5% discount.
R_2: Usually use the standard price.
$R_1 > R_2$

Here the priority $R_1 > R_2$ decides the conflict that arises in case both rules are applicable.

Of course other solutions are thinkable, in fact this problem is somehow solved in current systems. However, we hope to have illustrated the difficulties of monotonic rule systems. And if one wants to use a declarative approach, with all the benefits that follow for maintainability and formal analysis, then nonmonotonic rules are the way to go.

3 Description Logic $\mathcal{ALCNR}$

As an example of a description logic we consider the expressive language $\mathcal{ALCNR}$ [6] which formed also the basis for hybrid reasoning involving description logic and Datalog rules [16].

Concept and role descriptions are built from a set of primitive concept and role predicates (names), using a set of *constructors*. If C and D are concept descriptions, R a role description and A a primitive concept name, then the following are concept descriptions:

- A (primitive concept)
- $\top$ and $\bot$ (top, bottom)
- $C \sqcup D$ and $C \sqcap D$ (disjunction, conjunction)
- $\neg C$ (complement)
- $\forall R.C$ and $\exists R.C$ (universal and existential quantification)
- $(\geq nR)$ and $(\leq nR)$ (number restrictions).

Also, if $P_1, \ldots, P_m$ are primitive role names, then

$$P_1 \sqcap \ldots \sqcap P_m$$

is a role description.

A *terminology* T consist of

- *concept definitions* $A := D$, where A is a concept name and D a concept description. We assume that a concept name appears on the left hand side of at most one concept definition.
- *role definitions* $P := R$, where P is a role name and R a role description.
- *concept inclusions* $C \sqsubseteq D$, where C and D are concept descriptions.

The semantics of a terminology is given by *interpretations*. An interpretation I contains a non-empty domain $\mathcal{O}^I$. It assigns a unary relation $C^I \subseteq \mathcal{O}^I$ to every concept name in T, and a binary relation $R^I \subseteq \mathcal{O}^I \times \mathcal{O}^I$ to every role name in T.

An interpretation can be extended to concept and role descriptions as follows:

- $\top^I = \mathcal{O}^I$, $\bot^I = \emptyset$
- $(C \sqcup D)^I = C^I \cup D^I$, $(C \sqcap D)^I = C^I \cap D^I$
- $(\neg C)^I = \mathcal{O}^I - C^I$
- $(\forall R.C)^I = \{d \in \mathcal{O}^I \mid (d, e) \in R^I \Rightarrow e \in C^I$, for all $e\}$
- $(\exists R.C)^I = \{d \in \mathcal{O}^I \mid$ there is e with $(d, e) \in R^I$ and $e \in C^I\}$

- $(\geq nR)^I = \{d \in \mathcal{O}^I \mid \#\{e|(d,e) \in R^I\} \geq n\}$
- $(\leq nR)^I = \{d \in \mathcal{O}^I \mid \#\{e|(d,e) \in R^I\} \leq n\}$
- $(P_1 \sqcap \ldots \sqcap P_m)^I = P_1^I \cap \ldots \cap P_m^I$

Finally, an interpretation I assigns an element $a^I \in \mathcal{O}^I$ to every constant a.

An interpretation I is a model of a terminology T if $C^I \subseteq D^I$ for every inclusion $C \sqsubseteq D$ in T, $A^I = D^I$ for every concept definition $A := D$ in T, and $P^I = R^I$ for every role definition $P := R$ in T.

Let F be a set of facts of the form $A(a)$ and $P(a,b)$, where A is a concept name, P a role name, and a and b constants. I is a *model* of $A(a)$ and $P(a,b)$ iff $a^I \in A^I$ and $(a^I, b^I) \in P^I$, respectively. I is a model of F iff it is a model of every fact in F. I is a model of $F \cup T$ iff it is a model of T and a model of F.

A fact *follows* from $T \cup F$ iff every model of $T \cup F$ is also a model of the fact. We write $T \cup F \models A(a)$ and $T \cup F \models P(a,b)$, respectively.

4 Basic Ideas of Defeasible Reasoning

4.1 A Language for Defeasible Reasoning

A defeasible theory (a knowledge base in defeasible logic) consists of three different kinds of knowledge: strict rules, defeasible rules, and a superiority relation. (Fuller versions of defeasible logic also have facts and defeaters, but [1] shows that they can be simulated by the other ingredients).

Strict rules are rules in the classical sense: whenever the premises are indisputable (e.g. facts) then so is the conclusion. An example of a strict rule is “Professors are faculty members”. Written formally:

$$professor(X) \rightarrow faculty(X).$$

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is “Professors typically hold permanent positions”; written formally:

$$professor(X) \Rightarrow permanent(X).$$

The idea is that if we know that someone is a professor, then we may conclude that the person is a professor, *unless there is other, not inferior, evidence suggesting the contrary*.

The *superiority relation* among rules is used to define priorities among rules, that is, where one rule may override the conclusion of another rule. Rules are labelled to allow reference to them. For example, given the defeasible rules

$$\begin{array}{ll} r: & professor(X) \Rightarrow permanent(X) \\ r': & visiting(X) \Rightarrow \neg permanent(X) \end{array}$$

which contradict one another, no conclusive decision can be made about whether a visiting professor has a permanent appointment or not. But if we introduce a superiority relation $>$ with $r' > r$, with the intended meaning that r' is strictly stronger than r, then we can indeed conclude that the visiting professor does not have permanent appointment.

The relation $>$ is assumed to be acyclic. It can be declared to be transitive in certain (perhaps most) domains, but there is no need to do so for the formal system to work properly.

It is worth noting that, in defeasible logic, priorities are *local* in the following sense: Two rules are considered to be competing with one another only if they have complementary heads. Thus, since the superiority relation is used to resolve conflicts among competing rules, it is only used to compare rules with complementary heads; the information $r > r'$ for rules r, r' without complementary heads may be part of the superiority relation, but has no effect on the proof theory.

Now we briefly outline the proof theory. A *conclusion* of a defeasible theory D is a signed tagged literal. Conventionally there are two tags, and the signs $+$ and $-$. So a conclusion has one of the following four forms:

- $+\Delta q$, which is intended to mean that q is definitely provable in D.
- $-\Delta q$, which is intended to mean that we have proved that q is not definitely provable in D.
- $+\partial q$, which is intended to mean that q is defeasibly provable in D.
- $-\partial q$ which is intended to mean that we have proved that q is not defeasibly provable in D.

Provability is defined using inference conditions. Here we just describe the intuitive meaning of the inference condition for $+\partial$: To prove ∂L we need a rule r with head L which is applicable, that is all its antecedents a have been proven ($+\partial a$). In addition, we must counter all "attacks" on L: For every rule s with head $\sim L$ (the negation of L), (i) either there is an antecedent a which have been shown to be non-provable ($-\partial a$), or (ii) there is an applicable rule t with head L stronger than s: $t > s$. Thus each possible attack on the conclusion q must be counterattacked by a stronger rule.

5 Defeasible Reasoning Using Terminological Knowledge

5.1 Knowledge Bases

A *knowledge base* $K = (T, F, R, >)$ consists of

- A terminology T
- A set F of facts. Each fact has the form $p(a_1, \ldots, a_m)$, where p is a predicate, and $a_1, \ldots, a_m$ are constants. F is the disjoint union of a set F_T of facts with a concept or role predicate, and a remainder F_O. Predicates that are not concept and role predicates are called *ordinary*.
- A set R of rules, each of the form

$$L_1, \ldots, L_n \Rightarrow L$$

such that all L and L_i are literals $p(a_1, \ldots, a_m)$ or $\neg p(a_1, \ldots, a_m)$, with constants $a_1, \ldots, a_m$ and a concept, role or ordinary predicate p. Additionally, the predicate of L must be an ordinary predicate. $\{L_1, \ldots, L_n\}$ is the set of *antecedents* of the rule r, denoted $A(r)$. And L is called the *head* (or *consequent*) of r, denoted $C(r)$.
- an acyclic relation $>$ on R.

Now we make a number of remarks.

1. Rules with variables are interpreted as schemas: they represent the set of their ground instances. This interpretation is standard in many nonmonotonic reasoning approaches, among others in default logic [24] and defeasible logics [21].
2. The logical language does not have function symbols, thus the Herbrand universe is finite.
3. Concept and role predicates are not allowed to occur in the heads of rules. This design decision follows the idea that rules may not be used to derive ontological knowledge. All knowledge about concepts and roles is provided by the description logic component. The same idea was followed by other work on hybrid reasoning involving description logic and monotonic rules, e.g. [16]. The motivation for such an approach, and its relevance to the semantic web initiative, were outlined in the introduction.
4. Defeasible logic usually offers strict rules, and sometimes defeaters, in addition. We have omitted defeaters here, because they can be simulated by other means [1]. And we have decided to omit strict rules, because typically they include taxonomical, or other kinds of certain knowledge. We assume that such knowledge will be included in the ontology, and treated by the description logic. Instead we have allowed a set of facts about ordinary predicates. If need be, strict rules can be easily added to our logical system.

Given a set R of rules, $R[L]$ denotes the set of rules in R with head L. In the following $\sim L$ denotes the complement of L, that is, $\sim L$ is $\neg L$ if L is an atomic formula, and $\sim L$ is L' if L is $\neg L$.

5.2 Proof Theory

Given a rule r

$$L_1, \ldots, L_n \Rightarrow L$$

suppose $\{L_1, \ldots, L_n\}$ is partitioned into $\{L^{(1)}, \ldots, L^{(k)}\}$ and $\{L^{(k+1)}, \ldots, L^{(n)}\}$, such that the predicates of $\{L^{(1)}, \ldots, L^{(k)}\}$ are ordinary, and the predicates of $\{L^{(k+1)}, \ldots, L^{(n)}\}$ are concept or role predicates. Further suppose that $T \cup F \models L^{(j)}$ for all $j \in \{k + 1, \ldots, n\}$. Then the rule

$$L^{(1)} \ldots, L^{(k)} \Rightarrow L$$

is called the *reduct* of r; otherwise the reduct is undefined. For a set R of rules, $Red(R)$ collects the reducts of all rules in R.

A *derivation* (or *proof*) P is a finite sequence of signed tagged literals. $P(1..i)$ denotes the first i elements of this sequence. Now we proceed to give the inference conditions for $+\partial$ and $-\partial$. Let L be a literal with an ordinary predicate (Here we define ∂L only for such literals. A simple extension would be to derive $+\partial L$ iff $T \cup F \models L$, where the predicate of L is a concept or role predicate).

$+\partial$: If $P(i+1) = +\partial L$ then either
  (1) $L \in F$ or
  (2)   (2.1) $\exists r \in Red(R)[L]\ \forall L' \in A(r) : +\partial L' \in P(1..i)$ and
    (2.2) $\sim L \notin F$ and
    (2.3) $\forall s \in Red(R)[\sim L]$ either
      (2.3.1) $\exists L' \in A(s) : -\partial L' \in P(1..i)$ or
      (2.3.2) $\exists t \in Red(R)[L] \forall L' \in A(t)$
        $+\partial L' \in P(1..i)$ and $t > s$.

Let us illustrate this definition. To show that L is provable defeasibly we have two choices: (1) We show that L is a fact; or (2) we need to argue using rules. In particular, we require that there must be a rule reduct with head L which can be applied (2.1). But now we need to consider possible "counterattacks", that is, reasoning chains in support of $\sim L$. To be more specific: to prove L using rule reducts we must show that $\sim L$ is not a fact (2.2). Also (2.3) we must consider the set of all rule reducts which are not known to be inapplicable and which have head $\sim L$. Essentially each such rule s attacks the conclusion L. For L to be provable, each such rule s must have been established as non-applicable (2.3.1). Altenratively there must be an applicable rule reduct with head L stronger than the attacking rule (2.3.2).

$-\partial$: If $P(i+1) = -\partial L$ then
  (1) $L \notin F$ and
  (2)   (2.1) $\forall r \in Red(R)[L]\ \exists L' \in A(r) : -\partial L' \in P(1..i)$ or
    (2.2) $\sim L \in F$ or
    (2.3) $\exists s \in Red(R)[\sim L]$ such that
      (2.3.1) $\forall L' \in A(s) : +\partial L' \in P(1..i)$ and
      (2.3.2) $\forall t \in Red(R)[L]$ either
        $\exists L' \in A(t) : -\partial L' \in P(1..i)$ or not $t > s$.

Note that the inference condition $-\partial$ is the so-called strong negation of $+$ (de Morgan is applied, and $+$ and $-$ are interchanged).

Implementation will involve interleaving of description logic reasoners and defeasible reasoners. Derivability of antecedents with concept or role predicates will be checked by a description logic reasoner, the remainder is treated as specified in the inference conditions above.

Example 1. Imagine an online store which has organised its stock according to an ontology. Among others, the ontology contains the information

$$physicsBook \sqsubseteq scientificBook \sqsubseteq book$$

The pricing policy[1] of the store is written in defeasible logic, and might include the following information (scientific books get a special 5% discount).

$r_1 : \Rightarrow \neg discount(X, Y, Z)$
$r_2 :\ scientificBook(X) \Rightarrow discount(X, Y, 5\%)$

(where X denotes an article and Y a customer). The item with id 93215 is stored in the corporate data base as a physics book:

$physicsBook(93215)$

Putting all this information together, we can derive $+\partial discount(93215, Y, 5\%)$.

5.3 Ambiguity Propagating Defeasible Reasoning

We call a literal L *ambiguous* if there is a chain of reasoning that supports the conclusion that L is true, another that supports the conclusion that $\neg L$ is true, and the superiority relation does not resolve this conflict.

In [2] the property of ambiguity propagation was discussed, noting that the original defeasible logic was ambiguity blocking. A preference for ambiguity blocking or ambiguity propagating behaviour is one of the properties of non-monotonic inheritance nets over which intuitions can clash [27]. Ambiguity propagation results in fewer conclusions being drawn, which might make it preferable when the cost of an incorrect conclusion is high. For these reasons an ambiguity propagating logic is of interest.

The solution to achieve ambiguity propagation behaviour is to separate the invalidation of a counterargument from the derivation of $-\partial$ tagged literals. We do so by introducing a third level of provability (besides definite and defeasible provability), called *support* and denoted by $\int$. Intuitively, a literal L is supported if there is a chain of reasoning that would lead us to conclude L in the absence of conflicts.

$+\partial$: If $P(i+1) = +\partial_{am} L$ then either
(1) $L \in F$ or
(2) (2.1) $\exists r \in Red(R)[L]\ \forall L' \in A(r) : +\partial_{am} L' \in P(1..i)$ and
(2.2) $\sim L \notin F$ and
(2.3) $\forall s \in Red(R)[\sim L]$ either
(2.3.1) $\exists L' \in A(s) : -\int L' \in P(1..i)$or
(2.3.2) $\exists t \in Red(R)[L] \forall L' \in A(t)$
$+\partial_{am} L' \in P(1..i)$ and $t > s$.

Next we define the inference condition for support. ($-\int$ is defined accordingly, as the strong negation of $+\int$):

$+\int$: If $P(i+1) = +\int L$ then either
$L \in F$ or
$\exists r \in Red(R)[L]$ such that

[1] The idea of using discount policies as examples of declarative business rules is due to Grosof.

$\forall a \in A(r) : +\int a \in P(1..i)$, and
$\forall s \in Red(R)[\sim L]$ either
$\exists a \in A(s) : -\partial_{am} a \in P(1..i)$ or
$r > s$

Example 2. We continue Example 1. Suppose the discounting rules are modified, so that discount is only granted to "good" customers.

$r_1 : \Rightarrow \neg discount(X, Y, Z)$
$r_2 : \ scientificBook(X), goodCustomer(Y) \Rightarrow$
$discount(X, Y, 5\%)$

Rules for determining whether a customer is good can make use of corporate data bases where, among others, customer details and history data are stored. Rules governing the decision whether a customer is good or not might include the following:

$r_3 : \ boughtOverTwice(X) \Rightarrow goodCustomer(X)$
$r_4 : \ badPaymentHistory(X) \Rightarrow$
$\neg goodCustomer(X)$

Suppose a is a customer who has bought more than twice but has a bad payment history. If we don't use ambiguity propagation, then we can derive $-\partial goodCustomer(a)$. Then rule r_2 is blocked, and $+\partial\neg discount(93215, a, 5\%)$ can be derived.

However it is ambiguous whether a is a good customer or not. It may be the management's desire to let such cases be decided by a human, based on more details of the case (for example whether the recent payment history is good, or whether the customer buys big).

Ambiguity propagation allows this to happen. We can derive $-\partial_{am} goodCustomer(a)$, $-\partial_{am}\neg goodCustomer(a)$, $-\partial_{am} discount(93215, a, 5\%)$, and $-\partial_{am}\neg discount(93215, a, 5\%)$. The latter two conclusions might trigger human intervention, because the logic cannot make a decision (positive or negative).

We note that the situation would have been different if a stricter policy had been used, say, a customer with bad payment history should not be granted special discounts regardless of their purchase history. Formally we would specify $r_4 > r_3$. In this case $goodCustomer(a)$ is not ambiguous, and we can derive $+\partial_{am}\neg discount(93215, a, 5\%)$.

5.4 Properties and Relationships

First we show that all logics we have described above satisfy the basic property of *coherence*:

Theorem 1. *There is no knowledge base K and literal L such that $+\delta L$ and $-\delta L$ can be derived from K, where δ denotes any of the tags we have presented ($\Delta, \partial, \partial_{am}, \int$).*

Next we show an inconsistency can only be derived if already either the set of facts or the ontology is inconsistent.

Theorem 2. *Let K be a knowledge base and L an ordinary literal. If $L \notin F$ or $\sim L \notin F$, then we cannot derive both $+\partial L$ and $+\partial \sim L$, or both $+\partial_{am} L$ and $+\partial_{am} \sim L$.*

Finally we show that there exists a chain of increasing expressive power among several of the logics.

Theorem 3. $+\Delta \subset +\partial_{am,ntd} \subset +\partial_{am} \subset +\partial \subset +\int$.
For each inclusion there are defeasible theories in which the inclusion is strict.

6 Conclusion

This paper shows, for the first time, how description logics and defeasible reasoning can be combined. This kind of reasoning will find use in eCommerce applications on the Semantic Web. We defined a hybrid proof theory in which description logic reasoning and defeasible reasoning are interleaved. Also we gave some basic properties.

The approach taken in this paper is that ontological knowledge is only provided in description logic, and predicates defined in the ontology may be used in antecedents, but not in the heads of rules. If we allow concept and role predicates to occur in the heads of rules, the situation becomes more complicated. We intend to study such integration in the future.

References

1. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. Representation Results for Defeasible Logic. *ACM Transactions on Computational Logic* 2,2 (2001): 255–287.
2. G. Antoniou, D. Billington, G. Governatori and M.J. Maher. A flexible framework for defeasible logics. In *Proc. 17th American National Conference on Artificial Intelligence (AAAI-2000)*, 405–410.
3. G. Antoniou, D. Billington and M.J. Maher. On the analysis of regulations using defeasible rules. In *Proc. 32nd Hawaii International Conference on Systems Sciences*, 1999.
4. F. Baader and B. Hollunder. Embedding Defaults into Terminological Knowledge Representation Formalisms. *Journal of Automated Reasoning* 14,1 (1995): 149–180.
5. M. Balaban and A. Eyal. DFL – a dialog based integration of concept and rule reasoners. *Data and Knowledge Engineering* 38,3 (2001): 301–334.
6. M. Buchheit, F. Donini and A. Schaerf. Decidable Reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research* 1 (1993): 109138.
7. D. Connolly et al. *DAML+OIL (March 2001) Reference Description*. http://www.w3.org/TR/daml+oil-reference.
8. F. Donini, M. Lenzerini, D. Nardi and A. Schaerf. Reasoning in Description Logics. In G.Brewka (ed): *Principles of Knowledge Representation and Reasoning*, Studies in Logic, Language and Information, CLSI Publications 1996, 193–238.
9. F. Donini, M. Lenzerini, D. Nardi and A. Schaerf. A hybrid system with datalog and concept languages. In E. Ardizzone, S. Gaglio, F. Sorbello (eds): *Trends in Artificial Intelligence*, LNAI 549, Springer 1991, 8897.
10. A. Frisch (ed). *Workshop Notes of the AAAI Fall Symposium on Principles of Hybrid Reasoning*. AAAI Press 1991.
11. G. Governatori, A. ter Hofstede and P. Oaks. Defeasible Logic for Automated Negotiation. *Proc. Collecter'2000*.
12. B. Grosof. Prioritized conflict handling for logic programs. In *Proc. International Logic Programming Symposium*, MIT Press 1997, 197–211.

13. B. Grosof, Y. Lambrou and H. Chan. A Declarative Approach to Business Rules in Contracts: Courteous Logic Programs in XML. In *Proc. 1st ACM Conference on Electronic Commerce*, ACM 1999.
14. B. Grosof and T. Poon. Representing Agent Contracts with Exceptions using XML Rules, Ontologies, and Process Descriptions. In *Proc. ISWC'02 Rule ML Workshop*, Sardinia 2002.
15. I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-01)*, Morgan Kaufmann 2001, 199–204.
16. A. Levy and M-C. Rousset. CARIN: A Representation Language Combining Horn rules and Description Logics. *Artificial Intelligence* 104(1-2), 1998, 165 209.
17. M.J. Maher, A. Rock, G. Antoniou, D. Billington and T. Miller. Efficient Defeasible Reasoning Systems. *International Journal of Tools with Artificial Intelligence* (to appear).
18. M.J. Maher. Propositional Defeasible Logic has Linear Complexity. *Theory and Practice of Logic Programming*, 1,6 (2001): 691–711.
19. V. Marek and M. Truszczynski. *Nonmonotonic Reasoning – Context-Dependent Reasoning*. Springer 1993.
20. L. Morgenstern. Inheritance Comes of Age: Applying Nonmonotonic Techniques to Problems in Industry. *Artificial Intelligence*, 103(1998): 1–34.
21. D. Nute. Defeasible Logic. In D.M. Gabbay, C.J. Hogger and J.A. Robinson (eds.): *Handbook of Logic in Artificial Intelligence and Logic Programming Vol. 3,* Oxford University Press 1994, 353–395.
22. H. Prakken. *Logical Tools for Modelling Legal Argument: A Study of Defeasible Reasoning in Law.* Kluwer Academic Publishers 1997.
23. J. Quantz and V. Royer. A Preference Semantics for Defaults in Terminological Logics. In *Proc. 3rd International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann 1992, 294–305.
24. R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence* 13(1980): 81–132.
25. R. Rosati. Towards expressive KR systems integrating Datalog and description logics: preliminary report. In *Proc. 1999 International Workshop on Description Logics*, Linkping, Sweden 1999.
26. L.A. Stein. Resolving Ambiguity in Nonmonotonic Inheritance Hierarchies. *Artificial Intelligence* 55 (1992): 259–310.
27. D.D. Touretzky, J.F. Horty and R.H. Thomason. A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems. In *Proc. IJCAI-87*, 476–482, Morgan Kaufmann, 1987.
28. www.w3.org/2001/sw/