Nested Rules in Defeasible Logic

Insu Song and Guido Governatori

School of Information Technology & Electrical Engineering
The University of Queensland, Brisbane, QLD, 4072, Australia
{insu,guido}@itee.uq.edu.au

Abstract. Defeasible Logic is a rule-based non-monotonic logic with tractable reasoning services. In this paper we extend Defeasible Logic with nested rules. We consider a new Defeasible Logic, called DL^{ns}, where we allow one level of nested rules. A nested rule is a rule where the antecedent or the consequent of the rule are rules themselves. The inference conditions for DL^{ns} are based on reflection on the inference structures (rules) of the particular theory at hand. Accordingly DL^{ns} can be considered an amalgamated reflective system with implicit reflection mechanism. Finally we outline some possible applications of the logic.

1 Introduction

Nested rules arise naturally in our daily reasoning activities and in many applications: from artificial societies and normative reasoning, to configuration systems to security. Every time we have policies that are represented as sets of rules we have to consider the possibility that a policy contains rules about itself.

For instance, we often make decisions or classify objects based on some consequence relations. For example, in security, the usual definition of confidentiality is that a piece of information is regarded as confidential for an organization when the release of it would harm the interest of the organization. This can be formally written as:

$$(\mathit{Disclosed}(x) \Rightarrow \mathit{HarmInterest}) \Rightarrow \mathit{Confidential}(x).$$

In addition the security policy can give conditions (sometimes explicitly, sometime implicitly) about when the disclosure of a piece of information will harm the interest of the organization. In similar way many normative concepts frequently used in contracts, such as for example, delegation, empowerment, require definitions based on nested rules (see, for example, [1]).

In other cases, we often have rule dependencies that rule $r2$ is placed in a system only when rule $r1$ holds in the system:

$$r1 \rightarrow r2.$$

These dependencies are usually stored outside of the system. If rule dependencies can be expressed directly in the system, then $r2$ can be removed automatically whenever $r1$ does not hold in the system providing automatic system maintenance functionality. This feature is also useful in system integration because it allows context dependant rules.

A. Adi et al. (eds)
Rules and Rule Markup Languages for the Semantic Web. RuleML 2005.
LNCS 3791, pp. 204–208, 2005.

One major problem for adding nested rule expressions to any knowledge representation system is defining a proper evaluation of the nested rules. Evaluating the nested conditionals based on the material conditional fails miserably. The paradoxes associated with the material conditional are well known. For example, $(Disclosed(x) \supset HarmInterest)$ is logically equivalent to $(\neg Disclosed(x) \vee HarmInterest)$ so that the following statements are logically true:

1. If x is not disclosed x is confidential.
2. If interest is harmed by any reason, x is confidential.

In this paper we present DL^{ns}, which is a defeasible logic (DL) with nested rules and rule provability (see [2] for an introduction to defeasible logic). DL^{ns} allows one level of nesting of rules both in the antecedent and the consequent of non-monotonic statements.

The next section presents the proof theory of DL^{ns}. Then we show an example illustrating the use of nested rules. We conclude the paper with some remarks.

2 DL^{ns}: DL with Singly Nested Rules

In this section we outline a defeasible logic with singly nested rules (DL^{ns}) which admits one level of nesting of rules. A *nested rule* is a rule in the antecedent or the consequent of another rule. For example, $(a \rightarrow b) \rightarrow (c \rightarrow d)$ has two nested rules: $(a \rightarrow b)$ in the antecedent and $(c \rightarrow d)$ in the consequent. Rules in a DL^{ns} theory can contain nested rules, but nested rules cannot contain nested rules.

As in a standard defeasible logic (DL), a DL^{ns} theory is a triple $(F, R, \succ)$ where F is a set of literals (called facts), R is a finite set of rules, $\succ$ is a superiority relation on R. For the definitions of *literal* and *superiority relation* $\succ$, refer to [2]. A *relation* $r : (A(r), C(r))$ consists of its unique label r, its antecedent $A(r)$ which is a finite set of literals and nested rules, and its consequent $C(r)$ which is either a literal or a nested rule. A relation just says that $C(r)$ depends on $A(r)$. A *rule* $r_{\hookrightarrow}$ (i.e., $A(r) \hookrightarrow C(r)$) is a relation r with a rule type $\hookrightarrow$ specified. Replacing the placeholder $\hookrightarrow$ with the three rule types defined in DL yields three kinds of rules: $r_{\rightarrow}$ is a strict rule in the form of $A(r) \rightarrow C(r)$; $r_{\Rightarrow}$ is a defeasible rule in the form of $A(r) \Rightarrow C(r)$; $r_{\leadsto}$ is a defeater rule in the form of $A(r) \leadsto C(r)$. For example, a rule $(p \Rightarrow q)$ consists of its antecedent $A(r) = \{p\}$, its consequent $C(r) = q$, and its rule type $\Rightarrow$. A literal l is a strict rule with the antecedent the empty set and the consequent the literal itself: $\{\} \rightarrow l$. Given a rule $r_{\hookrightarrow}$, the *negation* of $r_{\hookrightarrow}$, $(N(r_{\hookrightarrow}))$ is the rule $(A(r) \Rightarrow {\sim}C(r))$. In Section 2.1 we will provide an intuition for this definition.

In DL, each type of rules represents a different strength of dependency between antecedents and consequents. We define *rule strength order* by which rules with the same relation can be ordered as follows:

$$r_{\rightarrow} > r_{\Rightarrow} > r_{\leadsto}$$

where $r_{\rightarrow}$ is a strict rule, $r_{\Rightarrow}$ is a defeasible rule, $r_{\Rightarrow}$ is a defeater, and r is a relation. The rules with the same rule types and relations have the same rule strength. Thus, rules with the same relation can be compared for their strength. For example, the following statements are true: $r_{\rightarrow} \geq r_{\Rightarrow}$, $r_{\Rightarrow} \geq r_{\Rightarrow}$, $r_{\rightarrow} \geq r_{\rightsquigarrow}$. Since a literal l is a strict rule $\{\} \rightarrow l$, the following statements are true as well: $l = (\{\} \rightarrow l)$, $l > (\{\} \Rightarrow l)$, $l > (\{\} \rightsquigarrow l)$.

The *final consequent* of a relation r is the right most consequent. For example, the final consequent of $a \rightarrow (b \rightarrow c)$ is c. If r is a literal, the final consequent of r is r itself. $A(r)_q$ is the union of antecedents in r for the consequent q. That is, $A(r)_q$ is the set of premises that need to be satisfied to conclude q. $A(r)_q$ is formally defined as follows:

$$A(r)_q = \begin{cases} A(r) & \text{If } C(r) = q \\ A(r) \cup A(C(r)) & \text{If } C(C(r)) = q \end{cases}$$

For example, given $r = a \rightarrow (d \rightarrow e)$, we have $A(r)_{(d \rightarrow e)} = \{a\}$ and $A(r)_e = \{a, d\}$.

Given a set R of rules, we denote the set of all strict rules in R by R_s and the set of strict rules and defeasible rules in R by R_{sd}. $R[q]$ denotes the set of rules in R of which q is either the consequent or the final consequent. For example, given R=$\{a \rightarrow (b \rightarrow c), d \rightarrow c\}$, we have $R[c]$=R and $R[b \rightarrow c]$=$\{a \rightarrow (b \rightarrow c)\}$.

$R[q]_{\hookrightarrow}$ is the set of rules satisfying the conditions of $R[q]$ and that all the rule strengths toward q are stronger than or equal to $\hookrightarrow$. For example: given R=$\{a \rightarrow (b \rightarrow c), a \rightarrow (b \Rightarrow c), a \rightarrow (b \rightsquigarrow c)\}$, we have $R[c]_{\rightarrow}$=$\{a \rightarrow (b \rightarrow c)\}$, $R[c]_{\Rightarrow}$=$\{a \rightarrow (b \rightarrow c), a \rightarrow (b \Rightarrow c)\}$, and $R[(b \rightsquigarrow c)]_{\rightarrow}$=$\{a \rightarrow (b \rightsquigarrow c)\}$.

2.1 Proof Theory

In order to make the presentation concise, in this paper we only consider DL^{ns} theories that R does not contain defeaters and rules with the empty set as their antecedent, such as $\{\} \rightarrow p$.

Unlike DL, a conclusion of a DL^{ns} theory is a tagged rule instead of just a tagged literal. Since a literal is considered a strict rule in DL^{ns}, this representation of conclusions includes tagged literals as well. The same set, $\{+\Delta, -\Delta, +\partial, -\partial\}$, of tags defined in DL is used in DL^{ns} with the exact same meaning.

In the course of derivations we will make use of auxiliary (sub) theories of a basic theory, and the elements of a derivation refer to these auxiliary (sub) theories. Thus given a theory D and a tagged literal $\pm\#q$, we use the notation $D(\pm\#q)$ to indicate that the tagged literal $\pm\#q$ has been derived/refers to the theory D.

Provability is defined below. It is based on the concept of a derivation (or proof) in $D = (F, R, \succ)$ as in DL. A derivation is a finite sequence $P = (P(1), \ldots, P(n))$ of tagged rules (or literals) satisfying the following conditions ($P(1..i)$ denotes the initial part of the sequence P of length i):

$+\Delta$: $P(i+1) = D(+\Delta q)$ if
(1) $\exists s \in R \cup F$ such that $s \geq q$ or
(2) $\exists t \geq q\ \exists s \in R_s[t]_{\rightarrow}\ \forall a \in A(s)_t$: $D(+\Delta a) \in P(1..i)$ or
(3) For $D' = (A(q), R_s, \emptyset)$, $D'(+\Delta C(q)) \in P(1..i)$.

To show that a rule (or a literal) q is definitely provable in D, i.e., $D(+\Delta q)$, we have three choices: (1) we show that a rule at least as strong as q is a rule of D; or (2) we show that a rule at least as strong as q can be deduced only from strict rules; or (3) we show that the consequent of q is provable definitely in the new theory D' consisting of the supposition (the antecedent of q) and the strict rules of D.

$+\partial$: $P(i+1) = D(+\partial q)$ if
(1) $D(+\Delta q) \in P(1..i)$ or
(2)(2.1) $\exists u \geq q\ \exists r \in R_{sd}[u]\ \forall a \in A(r)_u$:$D(+\partial a) \in P(1..i)$ and
(2.2) $D(-\Delta N(q)) \in P(1..i)$ and
(2.3) $\forall v \geq N(q)\ \forall s \in R[v]$ either
(2.3.1) $\exists a \in A(s)_v$: $D(-\partial a) \in P(1..i)$ or
(2.3.2) $\exists u \geq q\ \exists t \in R_{sd}[u]\ \forall a \in A(t)_u$: $D(+\partial a) \in P(1..i)$ and $t \succ s$ or
(3) For $D' = (A(q), R, \succ)$, $D'(+\partial C(q)) \in P(1..i)$.

To show that a rule (or a literal) q is defeasibly provable in D, i.e., $D(+\partial q)$, we have three choices: (1) we show that q is already definitely provable; or (2) we show that a rule at least as strong as q is defeasibly deduced from the defeasible part of D and that "attacks", which are reasoning chains in support of $N(q)$, are either not provable or defeated (i.e., they are weaker than appliccable rules for the conclusion we want to prove); or (3) we show that the consequent of q is defeasibly provable from an auxiliary theory D' consisting of the supposition (the antecedent of q) and all the rules of D. In (2.2) and (2.3), unlike DL, we consider reasoning chains in support of $(A(q) \Rightarrow \sim C(q))$ as "attacks" instead of $\sim q$. This is just one of the possible interpretations of a negation of a rule that has been considered in this paper. An argument for this is that one would be reluctant to accept $+\partial(a \rightarrow b)$ (and/or $+\partial(a \Rightarrow b)$) if any of the followings are supported: $+\Delta(a \rightarrow \sim b)$, $+\Delta(a \Rightarrow \sim b)$, $+\partial(a \rightarrow \sim b)$, or $+\partial(a \Rightarrow \sim b)$. Another possible interpretation of $N(r)$ is that the rule is not present in the theory. However, we do not pursue this interpretation here since it treats negation of literals and negation of rules differently.

The conditions for negative provability ($-\Delta$ and $-\partial$) can be constructed similarly following the principle of strong negation described in [2]. Thus, given the limited space, they are not presented here.

2.2 An Example

Let us consider the following scenario. A company has the policy that all confidential documents must be encrypted when they are sent by email, and no confidential document can be sent to people outside the company. A document is classified as confidential when its disclosure would harm the interests of the company. Let us suppose we have a document d describing the details of an application for a patent. Here we have that if the document is disclosed before the grant of the patent then the knowledge in it will be classified as public domain, and if something is public domain, other concurrent companies can use the technology described in the document. But if other companies use the technology, then its usage will generate less revenue than if it were secret and this will harm the interest of the company. This scenario can be described in a very natural fashion by the following DL^{ns} theory (in this example we use rule schemas, where each rule must be understood as the set of its ground instances):

r1: $(Disclose(x) \Rightarrow HarmInterests) \Rightarrow Confidential(x)$
r2: $Confidential(x) \Rightarrow Encrypt(x)$
r3: $Disclose(x) \Rightarrow PublicDomain(x)$
r4: $PublicDomain(x) \Rightarrow FreeUseOf(x)$
r5: $FreeUseOf(x) \Rightarrow LessRevenue(x)$
r6: $LessRevenue(x) \Rightarrow HarmInterests$

Now the question is whether a document describing a pending patent must be encrypted. To prove $Encrypt(d)$ we have to determine whether the document is classified as confidential. In this case we have to see whether we can prove the antecedent of the rule giving the condition to determine whether a document is confidential or not. Thus we have to use the rules in the theory to verify whether there is a relationship between the disclosure of the document and the potential harm caused to the interests of the company. In this case we assume hypothetically the $Disclose(x)$ holds and we try to derive $HarmInterst s$. This derivation succeeds and thus we can prove that the document must be encrypted.

3 Conclusion

We presented an extension of Defeasible Logic called DL^{ns}, which admits one level of nesting of rules both in the antecedent and the consequent of non-monotonic rules. It is constructed to demonstrate the general idea of our approach in developing DL^{n}, which accepts arbitrary nesting of rules such as $(a \rightarrow (b \Rightarrow c)) \rightarrow d$.

In the derivation of rules, our approach ensures appropriate connections between the antecedent and consequent of the rule as in relevant logic (see [3]) by insisting on relevance between antecedents and consequents by explicit rules being present in the theory for evaluating the rules.

We have introduced the concept that rules with the same relations can be ordered by their rule strengths (rule types). Using this concept and the requirement for an appropriate consequence connection between the antecedent and consequent of a rule, we have defined provability for both rules and literals. The provability condition is simple and it allows nested rule expressions and provides additional forms of conclusions such as $+\Delta r_{\rightarrow}$ and $+\partial r_{\Rightarrow}$.

References

1. Gelati, J., Governatori, G., Rotolo, A., Sartor, G.: Normative autonomy and normative co-ordination: Declarative power, representation, and mandate. Artificial Intelligence and Law **12(1-2)** (2004) 53–81
2. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: A flexible framework for defeasible logics. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, AAAI Press / The MIT Press (2000) 405–410
3. Anderson, A.E., Belnap, N.D.: Entailment: the Logic of Relevance and Necessity Vol 1. Princeton University Press (1975)