On the analysis of regulations using defeasible rules

G. Antoniou, D. Billington and M.J. Maher
CIT, Griffith University
Nathan, QLD 4111, Australia
db,ga,mjm@cit.gu.edu.au

Abstract

Regulations are a wide-spread and important part of government and business. They codify how products must be made and processes should be performed. Such regulations can be difficult to understand and apply. In an environment of growing complexity of, and change in, regulation, automated support for reasoning with regulations is becoming increasingly necessary. In this paper we claim that such automated support can be provided on the basis of defeasible logical rules. We highlight the support that can be provided by this logical tool, and illustrate some aspects using examples from one specific domain: university regulations.

Introduction

Regulations are a wide-spread and important part of government and business. They codify how products must be made and processes should be performed. Such regulations can be difficult to understand and apply. Even stand-alone regulations can be self-contradictory, as a result of the incremental process of their development and the lack of a formal drafting process. The problem becomes more difficult when independently developed regulations apply to a situation. For example, when two regulations overlap, it is not clear whether one regulation takes precedence or both regulations apply. Even when regulations are formally drafted, as is often done in the legal domain, problems with the consistency, interpretation and use of regulations still remain (e.g. (Sergot et al. 1986)). In an environment of growing complexity of, and change in, regulations, automated support for reasoning with regulations is becoming increasingly necessary.

In this paper we discuss issues of using logical methods to analyse regulations. In order to provide adequate support for the analysis of regulations a representation needs to meet some requirements:

- *Expressive power:* The method must be able to represent the basic features of the problem that needs to be modelled and analysed.
- *Reasoning support:* Not only should the features of the problem be representable, but it should also be able to manipulate the representations to derive properties of and conclusions from these representations. This lies at the heart of logical analysis.
- *Naturalness of expression:* It should be possible to represent the problem at hand in a transparent and natural way. Otherwise, even if the method satisfies the previous two requirements, it will fail, because it will be difficult to use the method in concrete situations.

We propose to use *defeasible rules* with *priorities* as a logical method of analysing regulations. Rules are normally sufficiently expressive to represent single items within a regulation. Since regulations may contradict one another the use of *defeasible* rules is adequate: they are rules that do not necessarily fire; rather they may be blocked by other rules with contrary conclusions. For more general arguments in favour of defeasible rules see (Schurz 1994).

Regulations commonly include exceptions. In addition there may be principles by which one set of regulations is superior to another (for example, a regulation of a higher authority is superior to a regulation of a lower authority; or a more recent regulation might override an older regulation). The priorities on defeasible rules are an adequate and natural way of representing this information.

The reasoning support will be provided by a particular logic we will be using: defeasible logic (Nute 1987, Nute 1994, Billington 1993). It is an approach to sceptical nonmonotonic reasoning (Antoniou 1997, Marek & Truszczynski 1993) that has a very distinctive feature: It was designed to be easily implementable right from the beginning, unlike most other approaches. In fact it has an implementation as a straightforward extension of Prolog (Covington 1997).

Defeasible logic is a sceptical formalism, meaning that it does not support contradictory conclusions. Instead it seeks to resolve differences. In cases where it is impossible to decide whether to conclude A or $\neg A$, the logic does not conclude either of them (thus the name "sceptical"). Sceptical reasoning is appropriate for the study of regulations. Users of regulations are mostly interested in getting correct advice without being confronted with conflicting views.

Basics of Defeasible Logic

A set of regulations will be represented as a defeasible theory. A defeasible theory[1] consists of four different kinds of knowledge: facts, strict rules, defeasible rules, and a superiority relation.

Facts denote simple pieces of information that are deemed to be true regardless of other knowledge items. A typical fact is that Tweety is a bird: $bird(tweety)$.

Strict rules are rules in the classical sense: whenever the premises of a rule are given, we are allowed to apply the rule and get a conclusion. When the premises are indisputable (e.g. facts) then so is the conclusion. An example of a strict rule is "Emus are birds". Written formally:

$$emu(X) \rightarrow bird(X).$$

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is "Birds typically fly"; written formally:

$$bird(X) \Rightarrow flies(X).$$

The idea is that if we know that something is a bird, then we may conclude that it flies, *unless there is other evidence suggesting that it may not fly.*

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

$$\begin{array}{lrl} r: & bird(X) & \Rightarrow flies(X) \\ r': & brokenWing(X) & \Rightarrow \neg flies(X) \end{array}$$

which contradict one another, no conclusive decision can be made about whether a bird with a broken wing can fly. But if we introduce a superiority relation $>$ with $r' > r$, then we can indeed conclude that it can't fly.

It turns out that we only need to define the superiority relation over rules with contradictory conclusions. Also notice that a cycle in the superiority relation is counterintuitive from the knowledge representation perspective. In the above example, it makes no sense to have both $r > r'$ and $r' > r$. Consequently, the defeasible logic we discuss requires an acyclic superiority relation.

[1] a knowledge base in Defeasible Logic

Examples: University regulations

Example 1: Academic misconduct

A typical rule found in university regulations is the following, taken from the Griffith University policy on academic misconduct:

> Where a student has been found guilty of academic misconduct on more than one occasion and has previously been penalised as set out in 3.1 - 3.3 above, the penalty shall normally be exclusion from the course, unless in the opinion of the relevant Assessment Board there are mitigating circumstances.

This is a typical *rule with exceptions.* In the framework we are proposing, we would represent this rule as follows:

r_1 : $guilty, repeat, previouslyPenalised \Rightarrow exclude$

$r_2 : mitigatingCircumstances \Rightarrow \neg exclude$

$r_2 > r_1$

Notice that the predicate *mitigatingCircumstances* will only be established if the Assessment Board decides so. Then a fact would be added to this particular case, and the decision would be not to exclude. This example illustrates the representation of exceptions in defeasible logic: Both the general rule and the exception are formalised as defeasible rules. The exception is stronger than the general rule (in case the exception rule is applicable, of course).

Example 2: Guidelines on fees

The second example is more substantial. It comprises part of the Griffith University guidelines on fees.

1.1 The University may not charge tuition fees for Australian students in undergraduate award courses.

1.2 The University may charge fees for postgraduate courses.

1.3 Overseas students are generally fee paying but there are some exceptions. There is minimum fee level set by the Government for fees for overseas students (FPOS). There are special arrangements for international exchange students. Refer to the GU International Center for advice on these issues.

1.7 All students are liable for HECS (Higher Education Contribution Scheme) with very few exceptions. Students who do not pay HECS include:

- FPOS students
- fee paying postgraduate students
- non-award students
- students with an APA (Australian postgraduate award)
- students wholly sponsored by an employer.

Here is the representation of this information in defeasible logic:

$r_{1.1}: student, australian, undergrad \Rightarrow \neg fee$

$r_{1.2}: student, postgrad \Rightarrow fee$

$r_{1.3a}: student, overseas \Rightarrow fee$

$r_{1.3b}: student, overseas \Rightarrow payFPOS$

$r_{1.3c}: student, overseas, exchange \Rightarrow \neg payFPOS$

$r_{1.7a}: student \Rightarrow HECS$

$r_{1.7b}: student, payFPOS \Rightarrow \neg HECS$

$r_{1.7c}: student, postgrad, fee \Rightarrow \neg HECS$

$r_{1.7d}: student, nonAward \Rightarrow \neg HECS$

$r_{1.7e}: student, APA \Rightarrow \neg HECS$

$r_{1.7f}: student, fullySponsored \Rightarrow \neg HECS$

$r_{1.3c} > r_{1.3b}$

$r_{1.7b} > r_{1.7a}$

$r_{1.7c} > r_{1.7a}$

$r_{1.7d} > r_{1.7a}$

$r_{1.7e} > r_{1.7a}$

$r_{1.7f} > r_{1.7a}$

Example 3: Hierarchies of regulations

Often regulations themselves are organised in a hierarchical fashion. For example on top of the Griffith University regulations there exist public service regulations which, if a conflict should arise, are stronger than university regulations.

Defeasible logic supports this structure in a natural way through its superiority relation: It is straightforward to encode the information that rules in a particular regulation are stronger than rules in another regulation.

How can formal methods support the analysis of regulations?

The use of formal representation and reasoning methods is beneficial for regulatory reasoning in various ways. In the following we distinguish between drafting regulations, and understanding and applying regulations.

Regarding the *understanding and application of regulations*, formal systems have the following advantages. These advantages are important, for example, for "naive users/subjects of regulation" who are regulated but do not wish to study the regulations.

1. *Decision support:* It is possible to run a specific case with the given regulations to get a correct answer.

2. *Explanation:* When an answer is given, there is also a reasoning chain explaining this response. This can be most helpful in help desks etc.

Being a logical approach, defeasbile logic provides this kind of support. *Drafting regulations* can be supported in the following ways:

3. *Anomaly detection:* Formal methods can be used to detect anomalies such as inconsistency, incompleteness and circularity. In defeasible logic, such anomalies are detected either by static analysis, or by the performance of the proof theory, for example by the proof of some facts regarding non-derivability (for technical details see (Antoniou et al. 1998)).

4. *Hypothetical reasoning:* It is possible to investigate the effects of changes to regulations on the entire regulatory system. This is possible since a defeasible knowledge base is an executable specification.

5. *Debugging:* In many cases we know what the answer to a specific query should be, yet the regulations in their current form lead to a different answer. Debugging suggest changes to the regulations which will have as an effect the desirable outcome. In defeasible logic, debugging can be carried out along the lines of "declarative debugging" (Naish 1997).

Conclusion

Regulations play an important role in the organisation and functioning of society in general, and business in particular. We pointed out that the increasing complexity of regulations and the frequency of necessary changes, mainly due to technological change and the current trend towards globalisation, make automated support in the analysis of regulations necessary. In this paper we argued in favour of a particular method, defeasible logic, which represents information in the form of rules and a priority relation. We used examples from university regulations to illustrate the adequacy of this representation.

Of course there is still a lot of work to be done in this area. One particular avenue we intend to explore in the near future is to build an argumentation support system for regulatory reasoning – a reasoning assistant.

This system will provide the analytical support outlined in this paper, and will do so in a user-friendly way.

Acknowledgements

This work is being supported by an ARC Large Research Grant.

References

G. Antoniou (1997). *Nonmonotonic Reasoning.* MIT Press.

G. Antoniou, D. Billington and M.J. Maher (1998). Normal Forms for Defeasible Logic. In *Proc. Joint International Conference and Symposium on Logic Programming* (accepted).

D. Billington (1993). Defeasible Logic is Stable. *Journal of Logic and Computation* 3: 370–400.

M.A. Covington, D. Nute and A. Vellino (1997). *Prolog Programming in Depth.* Prentice Hall.

V. Marek and M. Truszczynski (1993). *Nonmonotonic Logic*, Springer.

L. Naish (1997). A Declarative Debugging Scheme. *Journal of Functional and Logic Programming* 3.

D. Nute. Defeasible Reasoning (1987). In *Proc. 20th Hawaii International Conference on Systems Science*, IEEE Press, 470–477.

D. Nute (1994). 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, 353-395.

G. Schurz (1994). Nonmonotonic Reasoning and Changes of Belief. In P. Weingartner (Ed.), *Scientific and Religious Belief*, Special Issue of Philosophical Studies Series, volume 59, Kluwer, 47-64.

M.J. Sergot, F. Sadri, R.A. Kowalski, F. Kriwaczek, P. Hammond and H.T. Cory (1986). The British Nationality Act as a Logic Program. *Communications of the ACM* 29,5, 370-386.