Legal knowledge based systems

JURIX 91

Model-based legal reasoning

The Foundation for Legal Knowledge Systems

Editors:

J.A.P.J. Breuker

R.V. De Mulder

J.C. Hage

Nienke den Haan and Joost Breuker, A tractable juridical KBS for applying and teaching traffic regulations: J.A.P.J. Breuker, R.V. De Mulder, J.C. Hage (eds.), Legal knowledge based systems JURIX 91: Model-based legal reasoning , The Foundation for Legal Knowledge Systems, Lelystad: Koninklijke Vermande, pp. 5-16, 1991 ISBN 90 6040 989 2.

More information about the JURIX foundation and its activities can be obtained by contacting the JURIX secretariat:

Mr. C.N.J. de Vey Mestdagh
University of Groningen, Faculty of Law
Oude Kijk in 't Jatstraat 26
P.O. Box 716
9700 AS Groningen
Tel: +31 50 3635790/5433
Fax: +31 50 3635603
Email: sesam@rechten.rug.nl

http://jurix.bsk.utwente.nl/

A TRACTABLE JURIDICAL KBS* FOR APPLYING AND TEACHING TRAFFIC REGULATIONS

Nienke den Haan and Joost Breuker
Department of Computer Science and Law
Faculty of Law, University of Amsterdam
Kloveniersburgwal 72
1012 CZ Amsterdam
The Netherlands

Summary

In this article the solutions proposed by [Breuker and den Haan 91] concerning inferential explosions in the proof procedures in legal systems are put to practice. In our search for a tractable proof procedure we decided to use a formalism based on first order predicate logic to represent legislation without making use of deontic operators. The proof procedure we have developed also resolves the negation problem that emerges in AI and Law research. It appears that in this way legal reasoning can be reduced to simple matching and conflict resolution at the meta-level without the usual problems of modal operators and non-monotonicity. The Traffic Law domain serves as the testing ground for a prototype system.

1. Introduction

Let us start with establishing some terminology for this paper: with *article* we mean an enumerated section of a piece of legislation. The article is represented by a *rule* that is added to the regulation knowledge base. A certain piece of legislation handles a particular domain. The knowledge about the domain itself we call *world knowledge*. This knowledge contains only definitions of agents and actions from the domain itself and no normative notions.

The requirements for our proof procedure are that we do not only want to be able to determine the applicability of an article and establish its validation according to some situation, but also to perform legal reasoning with the articles. In this paper we propose several ways to support a tractable proof procedure. One part of the solution is the separation of pure normative knowledge and the commonsense world knowledge it refers to (see [Breuker and den Haan 91]). The separation of regulations and world knowledge is intuitively prudent, since it relates closely to the formulation of law. When world knowledge is separated from a rule, other rules may also refer to the same knowledge [1]. The articles assume knowledge of the underlying model about a particular commonsense world. The model cannot be constructed only on the basis of what is said about it in the articles, but preferably a domain expert should be consulted to make the model of the domain knowledge as complete as possible.

* Research supported by a grant from the Dutch Association for Scientifical Research into Traffic Safety (SWOV)

2. Goals of the Project

The report in this publication is derived from research conducted at the Department of Computer Science and Law for the RVV [2] project. The old regulations, the RVV66, were established in 1966 and have been valid until November 1st 1991. At that moment, the new RVV90 was brought into effect.

The lay-out of both laws is completely different. In the RVV60, rules were present for all traffic participants and almost each situation. This made the regulations quite voluminous. In the new version the drafters of the RVV90 have made successful attempts of describing more general and abstract situations, thus reducing the volume of the RVV. But consequently, comparing the meaning and contents of the old and the new version is not straightforward.

The two versions of the law gave rise to several research questions:

* Are there unintended contradictions?

* Are there unintended omissions?

* Is the new regulation complete?

* Is the new regulation coherent?

At the moment the drafters of the new regulation had to do all the checking by hand. Mismatches between the two versions were found relying only on human common sense and not by automatically exhaustive testing of computer-simulated traffic situations.

The SWOV expressed its needs to develop an environment in which traffic participants could be educated in traffic safety and safe behaviour on the road. Our group was asked to participate in this research. The general idea was that when the laws would be represented in a computer program, questions about the advisability of traffic situations could be answered or taught.

3. Theoretical Background

In [Breuker and den Haan 91] it is argued that although legal reasoning depends on deontic notions, in the case of regulation application the articles of law can be represented in a FOPL-formalism [3].

Current research in deontic logics does not provide a tractable proof procedure for deontic theories (see [Hilpinen 81]). However, AI-research has yielded proof procedures for FOPL. In order to establish a basis for our inference engine the deontic operators are moved out of the representation language and so the language in which the regulation knowledge will be represented is FOPL. The semantics of the rules representing the articles however still concerns legal necessity, i.e. when the conditions of a rule hold it is legally necessary that the conclusion will hold also, and it is not sensible to perform true/false validations on the regulation alone.

[McCarty 89] proposes a Language for Legal Discourse (LLD) which uses a mixed formalism of deontic modalities and world knowledge. The rules contain deontic predicates that also give conditions for the article: P(*phi*|*alpha*). [4] The description of the conditions contain all the necessary definitions of the world knowledge. Queries must ask specifically whether it is true that an action is either permitted, obligatory or forbidden under a particular condition *alpha*. Legal reasoning about *phi* with respect to (counter-) statements in other rules is impossible because in each rule the contents of the world knowledge is different and there is no way to make a good comparison without a commonsense world model of the domain. The article claims that proof procedures, restricted to Prolog-like inferences, become tractable for deontic modalities. This restriction is necessary because ``*deductive inference in a modal logic is notoriously difficult, but prototypical proofs are relatively simple*'' (see [McCarty 89, page 186], that suggests that there is a plausible account of such human commonsense reasoning in complex modal contexts).

Another viewpoint is to look upon the law as a logic program (cf. [Sergot 85] and [Sergot 86]). Although human commonsense reasoning is used to eliminate some negations in the rules, this direct translation of a piece of legislation means that the rules still contain world knowledge.Furthermore, information from a user is needed to supply numerous facts that can not be deduced automatically. The function of their program is to assess the nationality of a person; one cannot prove the consistency and coherence of the set of rules or the world knowledge, or experiment with either of them. They mention that they still have problems with negations because the embedded description of the world is not closed. In our approach we can opt for a complete description since knowledge engineers and domain experts can focus on the pure commonsense world. The need for such `deep models' was also recognised by [BenchCapon 89]. A complete description of the world containing several type-hierarchies, e.g. for several groups of agents and actions, limits the scope of a negation [5].

The separation of the pure regulation knowledge and the world knowledge it refers to gives the practical effect that attention can be given to the contents of the law and recognition of the underlying commonsense model of the domain. Just as articles employ abstract notions about the domain that are either explained in the definitions in the regulation or expected to be known by the reader, the rules contain references to the world knowledge. So we can either concentrateon the model of the commonsense world knowledge, or on the representation formalism of the articles.

3.1. Using Multiple Worlds

Worlds in which no laws are breached are juridically ideal worlds. Law texts are *prescriptive*, i.e. they prescribe correct behaviour. Basicly, the articles can be seen as prescriptive: forbidding, permitting or obliging. But at the same time they describe a world the legislators envisioned as being correct. Permitted or obligated actions can be seen in a juridically ideal world, but forbidden actions must be absent. Laws interpreted in this manner are describing an ideal world already.

When the legality of specific situations must be computed they are compared with the ideal world description. Real situations are excerpts of the actual world. Rules (either prohibitive or obligatory) of which the condition set fully matches the situation are used to give a *juridical validation*. The regulation itself is not validated, but the definitions of the rules are used (cf. forward chaining) to determine the legality of the *situation*.

In law texts seen as a juridically ideal world all rules must describe situations that are allowed. This implies that *articles* that *forbid* certain situations must be rewritten to *rules* that *allow the opposite*.

Let and *a,b,c* and *p* each be sentences which can be validated in the world knowledge. *O,F* and *P* express obligation, prohibition and permission.

Article	Rule
a, b, c -> Op	a, b, c -> p
a, b, c -> Fp	a, b, c -> ¬p
a, b, c -> Pp	a, b, c, p -> p

Figure 1: The translation of articles to rules

There is a problem when we want to translate permissions in the same way. The permissions do make an `ideal' statement in the sense of obligations, but they are more specific. The description of *p* must also be added to the set of conditions, since the rule must not fire when *p* does not occur under the circumstances *a, b, c*. [6] This solutions seems to be 'tautological', but one should keep in mind that these rules express juridical necessity and are not subject to truth-validations.

The obligatory articles are simply rewritten by omitting the deontic operator in the conclusions. In the articles about prohibitions the conclusions have to be *negated* [7] as well. The negations in the conclusions are as some may expect, no problem for the proof procedure, because the proof procedure uses forward chaining on the rules. To prove applicability we do not perform straightforward logical deduction on the rules; we only have to be able to match all the rule conditions to the description of the concerned actual situation.

3.2. Negations

The negated terms in prohibitive rules (*F*) can also be compared with the situation description. The type-hierarchies ensure that the scope of the rule is correct, e.g. the negation of a vehicle is never a type of road. Negations of types can only be taken from the same type-hierarchy (see figure2).

```
motorized_vehicle
        motorbike
        passenger_car
        bus
                service_bus
                tourist_coach
        lorry
```

Figure 1: Type tree of motorized vehicles

[Kowalski 89] tries to define a method to overcome negation problems by proposing transformations on rules with negations. The negations occur in several constructs over the articles: e.g. in the conditions of an article (e.g. $a,\neg b \rightarrow p$) that is an exception to another article (e.g. $a \rightarrow p$). By adding the condition without a negation to the counterpart rule, the negated condition in the exception rule may be removed (thus leaving $a,b \rightarrow p$). We think that there are side-effects because it is not guaranteed that the entire set of articles still agrees to the law text: altered condition sets may effect in an over- or underspecificity of the articles concerned.

This example is a description of a type-hierarchy of traffic agents (motorized vehicles). E.g. it can be established that a bus is a motorized-vehicle. When rules from the regulation base concern motorized vehicles, all the traffic participants mentioned in the type-tree on levels below are concerned.

The type-hierarchies as in figure 2 are also being used by a *situation generator*. In a situation there must be several agents, doing traffic actions on a type of road or crossing. The type-hierarchies provide restrictions for the situation generator on the types of concepts in a situation description.

4. The TRACS Prototype

The department is making the TRACS [8] prototype with the following short-term goals: the function of applying regulations, and a priority tool that selects the most important applicable rule. This module is called the EVALUATOR. Figure 3 shows its architecture.

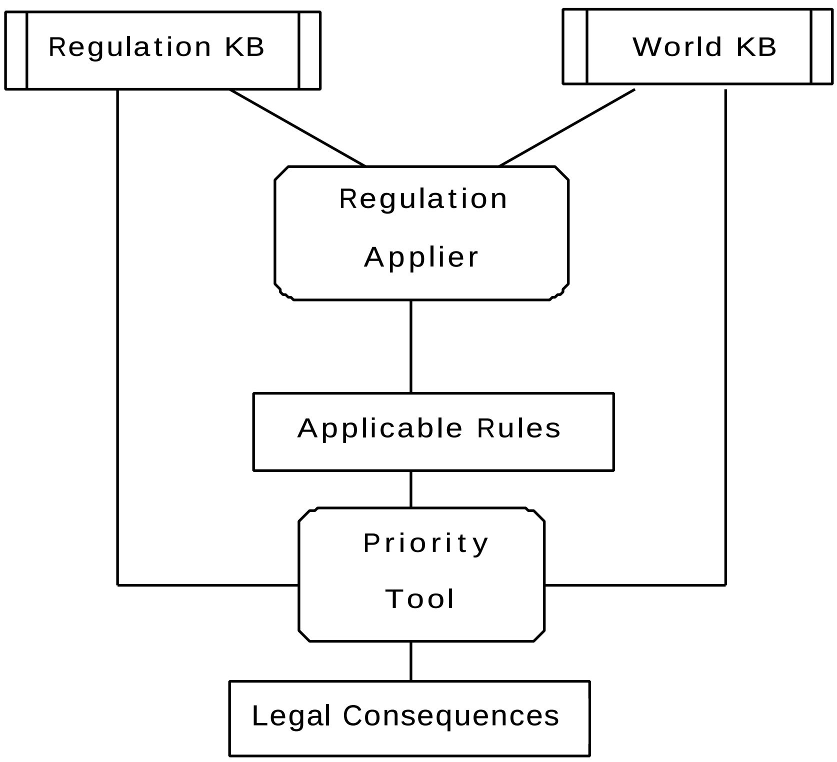

Figure 3: Architecture of the Evaluator

4.1. Regulation Knowledge Base

The domain knowledge is divided into two main parts. [Breuker and den Haan 91] explains the division of regulation knowledge and commonsense knowledge. The regulation text contains references to the commonsense knowledge, stored in the world KB.

All $x_{1..n}(F_1(x_1)\underline{op}..\underline{op}F_n(x_n)) \rightarrow G(y)$

All the regulation knowledge is universally quantified since the regulations are meant to be valid in all situations for all traffic participants. In this formula the F_i denote references to world knowledge (relations between variables or other predicates). The operators (op) may be either conjunctive ($\wedge$) or disjunctive ($\vee$). The X_i are typed variables, e.g. V:vehicle. Frequently, they occur in G(y): $F_{1..n}$ describe the necessary conditions for an article and G(y) contains the juridical statement about the agent(s) in question.

4.2. World Knowledge Base

Determining the applicability of a rule is no longer purely legal reasoning: the core of the proof procedure has been moved to the commonsense knowledge (world KB). The rules in the regulation KB are truth-validated according to the current situation description. Since all the terms in the rule refer to world knowledge, the terms are compared using many-sorted resolution.

To determine the degree of applicability of a rule, the semantic distances of the terms in the rule to the terms in the situation is computed using the type-hierarchies (see Frisch:91 on many-sorted unification). E.g. when reviewing a situation with a tourist coach, articles about buses are more important than articles about motorized vehicles. Both apply, but rules containing agents of types of the level closest to those of the agents in the current situation have the highest priority.

The world knowledge in the traffic world is strongly typed and is organized in type-hierarchies such as figure 2. So, the many-sorted resolution could be based on proving the intercorrelative subsumption relation: a variable can be matched to another if its type is equal or a lineal descendant of the other type in the type-hierarchy.

Negations can also be proved using type-hierarchy: types that have a common root and are no lineal descendants are contradictory. [9]

4.3. Regulation Applier

The EVALUATOR in the first version tests all the rules in the regulation base consecutively. In later versions the user may choose to test certain paragraphs of law texts, or only certain aspects of a situation (agents or actions, traffic rules or the traffic signs and lights as well).

In each article the grounds for its applicability are described, i.e. a rule may concern a certain type of traffic actions or agents. E.g. vehicles must give way at a crossing to vehicles coming from the right (article 16.1). When the condition part of a rule matches the situation description, the semantic distances of the terms of variables in the situation and the types in the rule are also stored.

Consider the following example:

* Art 3: It is obligatory that vehicles keep the rightmost position on the road.

* Art 5.1: It is obligatory that bicycles use the bicycle path.

The representation of the articles results in the following rules:

* rightmost(X:vehicle, R:road)

* road position(X:bicycle, B:bicycle path)

Let the actual situation consist of a bicycle on the bicycle path which is not the rightmost lane of the road. Article 3 applies since the type in the situation description (bicycle) matches in the type-tree to a vehicle. In article 5 the types match exactly, since the types of the variable in the rule condition and the actual situation description are equal.

Only after testing the conditions, the prescriptive part needs to be checked. For juridically correct behaviour in the actual world the conclusion parts of the applicable rules must also match the situation description. In the example, the conclusion of article 3 is not matching the actual situation description, so article 3 is breached. The conclusion part of article 5.1 matches to the situation description, so article 5.1 yields a positive match.

In this way, looking for applicable rules is reduced to simply comparing the situation description and the condition part of rules. The subsequent testing of the legal conclusions is done similarly; this time the situation description is compared with the conclusion.

The number of rules is finite, as well as the number of references to the world knowledge. Furthermore, to establish the applicability of a rule to a description of a situation of the actual world no complex proofs are needed: the truth of the conditions in terms of the situation descriptors must be proved. The conditions refer to the world description, so the only computation problem could lay within the world knowledge.

4.4. Priority Tool

To make a further selection on the set of applicable rules a juridically correct choice must be made. The priority tool provides an ordering over the applicable rules (for more on metaprogramming, see [Hamfelt and Barklund 90]). Hamfelt and Barklund recognise that legal knowledge is partitioned in several levels. In this approach the proof procedure follows these levels as consecutive steps: first applicable rules are selected and then the priorities are determined.

The priority relations are determined for the sets of rules yielded by the regulation applier. The rules in the regulation themselves are not ordered in any way, since the ordering is different for each situation. They concern instantiated, specific situations, since the semantic distances are used to select the most important rules.

In the prototype, the priority tool of the Evaluator consists of just one rule that allows one to compute the semantic distances of rules to situations. This priority rule is based on the *lex specialis legi generali derogat* paradigm, i.e. specific articles overrule more general articles:

* The semantic distances between all terms in rules and situations are computed totaled up in order to determine the measure of generality of the entire article.

* The generality of a rule is also reflected in the number of its conditions. The condition sets of the rules are compared: a superset is more restricted and thus more specific.

As soon as the rules concerning traffic signs and lights are included in the regulation KB, they will impose the need for extra priority rules. The RVV90 explicitly states that signs have a higher priority than ules, and that lights have a higher priority than signs (articles 62 and 63). Domain experts from the SWOV explained that another priority rule, which is implicit,is based on the order of the articles in the law-text. The RVV90 is structured in such manner, that (within a paragraph) rticles with higher numbers have a higher priority.

In the example, the match of the variable in article 5.1 is closer than he match of article 3 because article 5.1 specifically mentions bicycles. t turns out that the positive match of article 5.1 has a higher priority than the negative match of article 3. The conclusion is that no article is breached. In this simple case, the meta-rules were able to narrow down the ets of applicable rules to one single article.

4.5. Visualization

In the prototype the traffic situations will be displayed as follows. irst the initial situation is shown. In this situation one or two roads are displayed, and simple drawings depicting traffic participants. Then traffic actions are shown, either verbally or in the picture. In the prototype displayed movements of vehicles are shown in three phases and are not smoothly animated.

The traffic actions lead to the final situation which is also displayed on the screen. In this first rigid version at least the traffic actions and agents are visible to the user, but only in later editions the user will be able to *interact* with traffic situations. Traffic agents may be moved in the situation on the screen by picking them up with the mouse and moving then around in order to perform traffic actions. These movements in the situation, either user-imposed or computer-generated, will be shown as animations.

5. Further Research

Crafting a tutoring and advisory system for Traffic Law fits into the research

programme of the LRI. The global goal is to develop a workbench for legal practitioners (see [Breuker 90]).
This effort concerns the following extensions:

* Support of law research:
law comparison: between different versions or between countries
law development: our representation formalism allows to experiment with changes in either the regulations or in the world knowledge.

* By using a generator and the world knowledge, all possible and relevant situations can be generated. In this way (elying on the AI-practice that ensures a coherent description of the commonsense world knowledge) the coherence and completeness of the laws can be examined, as well as the equivalence of laws.

* By backward chaining on the rules an effort can be made towards proving the consistency of a law: when the conclusions of two rules are opposite then the condition sets must be different.

* Planning ligitation The interest of a client is best served when no articles which could make a negative match with a possible situation will be applicable. his may mean that when a certain article is bound to be breached, a counterarticle is searched in the regulation knowledge base. The set of conditions and the types of the variables in the condition set of the counterarticle are investigated in order to determine in which way a client should act to make that positive counterarticle applicable.

The tutoring tool that will be added to the TRACS system will exploit the eparate representation of regulation and world knowledge. uring experimentation on the definitions of regulations and their ossible implications, one may choose to change the rules, e.g. in the process of law design and development, or the world, i.e. to check whether the current set of rules would still be adequate and lead to satisfactory deductions when the actual world offers new actions. An example of a change in the world is the following. In the beginning of this century the case emerged of a man who tapped electricity from his neighbours. The old laws did not recognise electricity as a good that could thus be stolen. Ofcourse what was necessary was a law that recognised electricity as a good, so the essential article about goods was extended. A sequel to this case was the accused theft of electronic information (e.g. computer programs, registration databases, etc.). This also led to more research and eventually the reformulation of the relevant articles.

6. Conclusions

The representation formalism developed during this research and the system module that tests rules for their applicability forms the basis for extending the prototype. Tutoring and giving advise are both dependent on the rule evaluator.

The solution of the tractability problem is twofold: we make a separation of the regulation knowledge (rules) and the world knowledge, and we use normal FOPL-based proof procedures with many sorted unification. We used the notion that regulations give a description of a juridically ideal world: we are able to use true false validations on world knowledge as a sufficient basis for the proof procedure. The deontic operators are kept out of the representation of the articles. The computational aspects of the world knowledge can be estimated since AI-experience with representing various domains provides some insight into the depth of the proof procedure. [10]

The complete Traffic Regulation Animation and Comparison System will consist of:

* Situation Generator This is a very simple tool that uses the type-hierarchies as input and exhaustively generates traffic situations.

* Dialogue and Explanation Facilities The user is enabled to ask direct queries or select generated situations concerning the applicability of articles. The behaviour of the problem solver can also be explained in terms of semantic closeness or priority rules.

* Interactive User Interface The user is enabled to define or choose interesting traffic situations and to impose traffic actions.

* Intelligent Tutoring Intelligent tutoring systems allow law students to ask questions, learn the tricks of the legal trade and experiment freely with the regulation and world knowledge bases. In the IT-tool the model of the student's knowledge of the regulations and validations is updated according to her/his progress. Misconceptions of regulations are met by comparing the student model with the rules from the regulation KB. Mistakes in juridical problem solving are either on the priority-rules matters or are spotted when the student-model is compared to an expert-model. Expert problem solving models are acquired by having lawyers or other legal experts tackle traffic law problems.

* Consistency Checker By using backward chaining inconsistent rules can be spotted.

* Comparing old/new versions of laws The situation generator exhaustively tries situations on both the RVV66 and the RVV90. The conclusions should be similar in each situation. Only then both versions can be said to have equal effects.

When the tools developed in the course of this research will have been embedded in a toolkit, the potentials of such a workbench are high. The system will aid lawyers and sollicitors in the ligitation process. It will form an overall support, the expert user can easily try out her/his theories.

7. Acknowledgement

Many thanks to Manfred Aben for trouble-shooting the theories and proof-reading this article.

8. Notes

1. According to our national background we occupy ourselves with the western-continental law system which is highly statute-based as opposed to the anglo-american system which is mostly based on case-law.
2. Reglement Verkeersregels en Verkeerstekens: The Traffic Regulations.
3. First Order Predicate Logic.
4. Dyadic deontics: *phi* is permitted under the circumstances (conditions) *alpha*.
5. Otherwise, the opposite of a vehicle might be a pedestrian crossing.
6. The same problem occurs in the standard AI-problem of the penguin. Although the animal has all the forecomings of a bird, it cannot fly.
7. Many negative statements in regulations are in the conclusion parts of prohibitions. Following our method the conclusions expressing prohibitions are negated again, so actually we loose numerous negations.
8. Traffic Regulation Animation and Comparison System.
9. In a type hierarchy sibling nodes are mutually exclusive.
10. The rules only have a linear influence, since there is a finite set of rules in a law text.

8. References

T.J.M. Bench-Capon (1989). *Deep models, normative reasoning and legal expert systems*. In Proceedings of the 2nd International Conference on AI and Law, Vancouver. ACM.

Joost Breuker and Nienke den Haan (1991). *Separating world and regulation knowledge: where is the logic?* In Proceedings of the third international conference on AI and Law, Oxford.

Joost Breuker (1990). *Towards a Workbench for the Legal Practitioner.* In Legal Knowledge Based Systems: Aims for Research and Development. Proceedings of JURIX Conference, Leiden.

Joost Breuker (1991). *Artificial Intelligence and Common Sense in Law.* Inaugural Speech.

A.M. Frisch (1991). *The substitutional framework for sorted deduction: fundamental results on hybrid reasoning.* Artificial Intelligence, 49(1-3):161-198.

A. Hamfelt and J. Barklund (1990). *Metaprogramming for representation of* *legal principles.* In Proceedings Meta90, pages 105-122. Katholieke Universiteit Leuven.

R. Hilpinen, editor (1981). *Deontic Logic: Introductory and Systematic Reading*. Reprinted, first published in 1971.

R.A. Kowalski (1989). *The treatment of negation in logic programs for representing legislation*. In Proceedings of the 2nd International Conference on AI and Law, Vancouver, ACM.

L.T. McCarty (1989). *A Language for Legal Discourse I. Basic Structures.* In Proceedings of the 2nd International Conference on AI and Law, Vancouver, ACM.

M.J. Sergot (1985). *Representing Legislation as Logic Programs.* In J.E. Hayes, D. Michie, and J. Richards, editors, Machine Intelligence 11, pages 209-260. Oxford University Press.

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.* In Communications of the ACM 29,5, pages 370-386.