Computational Models of Large Scale Systems of Legal Rules

B. Intrigila*, M. Cianci*, P. Damiani*
P. Di Salvatore**, A. Centanni***

KEYWORDS: Computational models in the legal field, fuzzy logic and fuzzy reasoning.

ABSTRACT: We discuss computational models of systems of legal rules and, more generally, the computational approach in the legal field. In particular, we point out a set of basic requirements to be fulfilled by formal models of legal rules. Next, we treat the use of fuzzy logic and fuzzy reasoning to set up suitable models to be used in computer simulations. The work is accomplished in a general setting, so that it could be re-used also for other experimentations.

INTRODUCTION

In the present paper, we consider computational models of systems of legal rules, that is «rules» (suitably represented) which are, in a sense, the abstraction of existing legal rules, resembling them in many relevant aspects, e.g. in shape (see later, Section 2). The exposition is organized as follows.

Section 1 is devoted to give the main motivations for constructing such kind of models, and to expose the general setting in which this task is here accomplished.

In Section 2, we discuss some requirements on formal models of legal rules. Such requirements come from the judicial theory, which points out the formal structure of norms, and also from the computer science needs,

* Dipartimento di Matematica Pura ed Applicata - Università di L’Aquila - Via Vetoio - 67010 Coppito (L’Aquila) - Italy.
** Istituto di Teoria dell’Interpretazione ed Informatica Giuridica - Università «La Sapienza», Roma.
*** Ministero di Grazia e Giustizia: Direzione Generale per l’Organizzazione Giudiziaria.
since we are looking for computationally treatable models, which determines a suitable representation of legal rules.

Section 3 illustrates the use of Fuzzy Logic in our computational model; moreover, to facilitate the unexperienced reader, we have included some basic notions concerning the use of Fuzzy Logic in control systems. This Section ends with some worked examples.

Section 4 describes the future work to be done to achieve the task of building up complete computational models, and in particular the general tool we are setting up to automatically generate the kernel of a computational system of rules.

1. THE NEED FOR COMPUTATIONAL MODELS

Here we consider legal systems in the most wide sense, that is any information system, composed by rules, on which base some specific case gives rise to some legal decision. So, we are adopting a Normative Model, according to the classification of [GS1] (Chapter III.1). By a legal decision, we mean the assignment of some deontic value; we prefer here to relay on the intuitive meaning of «legal decision» rather than commit ourselves to any specific formal system of deontic logic.

1.1. Interaction Problems

Our starting point is the fact that interaction with existing legal systems is a difficult task [HT]. Moreover, the construction of computer-based tools for significantly improve such interactions is difficult too.

There is a general agreement on the need of a better understanding of such difficulties. To be more concrete, let us enlist some of them, without any claim to be exhaustive or complete:

(i) at a logical level we find that it is not clear how far the existing formal systems of (deontic) logic are able to correctly represent or simulate the logical inference mechanisms of actual legal decisions; in saying this, we are not referring to paradoxes of deontic logic (which are, in any case, of great importance), rather, to practical problems in modelling valid legal inferences [GS1, GS2, McC, DT, AV];

(ii) a related problem is the representation of legal rules, in this respect classical logic seems not to be suitable as a language to formally represent legal knowledge, since we have often to encode incomplete, uncertain or even contradictory information; classical logic (and therefore logic pro-
gramming) needs to eliminate uncertainty, so making a sequence of doubtful choices, at least enforcing sharp interpretations and discarding other acceptable ones (e.g. [CB]);

(iii) the elicitation of legal rules since actual rules are embedded in texts, written in natural languages, in a much more complex way than, say, mathematical propositions are embedded in mathematical books or articles; moreover according to [GS2] legal language, though partially technical, borrows the terminology of the field it intends to regulate. Therefore to accomplish the elicitation task we must often refer to common sense, and this is not at all a mechanical step. «In this respect legislation can be viewed as programs expressed in human language to be executed by humans rather than computer» [K].

(iv) moreover we find the interpretation problems, since, even if we assume that the extraction of the rules from the texts has been made in successful way, we are faced with ambiguities not depending on natural language, but related to the very structure of the rules themselves («Like ambiguity, vagueness is also a form of imprecision in the law, but one that is usually intended. A draftsman will often make a conscious decision to leave some concepts undefined and vague, because he prefers that meaning of the concepts should be determined later in the context of real cases when all the individual circumstances can be given proper consideration. A draftsman could never anticipate all the possible combinations of circumstance that might arise in the future and make explicit provision for them. Vagueness in the law is essential. It enables the law to adapt to unanticipated circumstances and to adjust gracefully to changing needs» [KS]);

(v) as in most cases, such ambiguities are to be solved referring to other legal rules, we are led to the complex system of relationships between legal rules, which includes cross-references, hierarchical and temporal relations, etc. [PDGMM];

(vi) as far as we want to apply rules to specific cases, we have analogous problems: case representation formalisms, relevance of a rule to a specific case, finding all relevant rules, etc.

1.2. Computer-based Interaction Tools

As previously stated, in the construction of such tools a number of problems have not received a completely satisfactory solution. We can distinguish, in order of difficulty, the following application fields:

- Information Retrieval [HT];
- Computer Aided Search and Navigation in Legal Databases [PGDMM];
– Computer Aided Decision Making;
– Legal Drafting [BMS];
– Expert Systems [AV].

We limit ourselves to discuss this last item, where all problems are to be faced.

First of all, we remark that an enormous work has be done on constructing formal logical systems, to represent, at a more or less high level of abstraction, the inferential legal mechanisms; also much work has been done in representing legal knowledge as well as the conditions of applicability of legal norms or rules to specific facts or actions.

If we have:

- Representation of Knowledge
- Representation of Facts
- Inferential Mechanism

then, as it is well known, we are in position to obtain an expert system, that is a Knowledge Data Base, which acts on (representations of) Facts, via the Inferential Mechanism.

Despite some successes, it is commonly recognized that the realization of a completely satisfactory Expert System in some law field is a tremendously difficult task, which can be partially accomplished only in very restricted domains. The reason of this difficulty can be found in the mentioned phenomena, which in an expert system perspective act as follows.

Considering, for instance, the Knowledge Elicitation process, we have to put the legal rules in the chosen representation system, mainly extracting them from texts. Here we have the following dilemma:

(a) an automatic treatment of texts to this aim, is out of the scope of actual Natural Language Processing techniques;
(b) on the other hand, if we use human experts to do the job, we then find two main problems:
- high costs for little benefits, since legal rules are rapidly changing
  «U.S. case law comprises roughly 50 gigabytes of text and grows by 2 gigabytes per year» [HT];
- high risks to commit the system with some subjective interpretations of the law.

1.3. Mathematical and Statistical Models

While such difficulties are well known, what seems to be still lacking is a suitable modelling of such phenomena. Put in different words, we would
like to have formally precise models which represent the various forms of interaction between legal systems and the users (of different kinds) of these systems.

To this aim, we may ask what kind of approach is suitable to improve our comprehension:

(A) the theoretical, i.e. mathematical, approach;
(B) the empirical, i.e. statistical, approach;
(C) the computational, i.e. computer simulation, approach.

As an example in the Information Retrieval field [smc], the construction of more effective tools to interface text databases strongly relies on mathematical and statistical modelling of phenomena such as occurrences of words or combinations of words in texts. So, we can often give a proper assessment of what happens when a database user formulates a query of such a form. This corresponds to the abovementioned approach A.

Since this is not completely reliable, we need an evaluation of empirical performance of existing IR tools on collections of texts or documents specifically built up as test benches, as the TREC Test Collection.[trec] This corresponds to approach B.

Coming back to the Legal Field, the following considerations are in order.

(A) It is well known that there are not reliable models of the abovementioned interactions between users and legal systems.

(B) Moreover, it seems not convenient to cope with such problems via extensive statistical investigations.

In fact:

1) it is likely that any conclusion obtained by investigating a specific field of law, cannot be applied to other fields without introducing (large) interpretations errors of (desperately) difficult assessment;

2) a pool of experts is required to evaluate the interactions users-system, and the results will be strongly biased by the choices of experts;

3) the enormous number of relevant norms requires a very large (and consequently very expensive) sampling; and in many cases a reliable sampling would be exceeding large;

4) the legal knowledge quickly changes as new laws, statutes, regulations supersede the previous ones. Therefore it is very difficult to evaluate how far our results will remain valid in the next future.
The previous facts seem to indicate that this approach, undoubtedly useful, is, however, of difficult application in the Legal Field.

(C) When approaches (A), (B) are not successful, the scientific research indicates that computational models are valuable investigations tools.

1.4. Computational Models

Computational models needs a modelling of the phenomena under study. Once this is done, we can use them in massive computer simulations.

It is obvious that the results so obtained are strongly related to the model choices, and therefore they are reliable as far as the model is. On the other hand, even if the characteristics of the model are questionable or doubtful we can obtain interesting results, at least for an analysis of the choices behind the model's construction.

So, what is attractive in the Computational Approach, is the possibility of a continuous refinement of models \textit{via} the simulation process, without high costs. This seems to be valuable in the Legal Field:

- to obtain a better comprehension of the above-mentioned interaction problems;
- to set up test benches of possible or existing tools;
- to «experiment» with new theoretical ideas.

We just remark that Computational Models, when reasonably near to reality, can be useful for many different tasks. So, for example:

«An engineer who is designing a new bridge will construct a complete model to test out various aspects of design. There will always be aspects of the design that are impossible to test. But no engineer would complete actually building the bridge until all the possible tests had been completed. We believe that there is now at least the possibility of making similar computer models available to a legal draftsman.

Such computer models for the draftsman need not to be limited to simulating the effect of legislation on a library of stereotypical cases. Mechanical theorem provers might be used to derive logical consequences from a logical model representing the current or proposed new draft of a piece of legislation. One use of this capability might be to test whether the legislation does possess intended general properties. A failure to establish that such a general property hold would identify mistakes or gaps in the legislation, suggesting in turn how the defective draft might be improved.» [ks].
1.5. Computational Legal Rules

Our starting point has been the need to experiment with some new ideas for interacting with textual legal databases. We wanted to evaluate the feasibility of such ideas without actually implementing them. So, we decided to set up a computational model to accomplish such evaluation. Here we briefly describe this setting, which will be explained in details in the Section 2.

1.5.1. The Kernel of the System

(1) The basis of the model is the legal knowledge; we have represented it as a set of production rules, which can be applied, via an inferential mechanism, based on fuzzy logic, to get «legal judgements» about specific «cases».

To set up a concrete experimentation, all rules refer to a simplified «Highway Code».

(2) Also using fuzzy logic, we have modelled the interpretation of legal rules, that is the phenomenon which produces different judgements for the same case, because of a different «subjective interpretation» of the same rules.

1.5.2. The Fuzzy Logic Inferential Tool

(1) The inferential mechanism associated to the production rules is constructed using a (slight) modification of the DUC (Dublin City University) - Fuzzy Tool [YRP].

This tool has been devised to build expert systems to be used (among other applications) in control of industrial processes, similar to the existing ones, such as Blast Furnace Control, Automotive Speed Control, Rainwater Pump Management, etc.

Therefore the inferential mechanism is highly reliable, flexible and easy to modify.

(2) The tool is easy to use, since it amounts to writing down:
- the linguistic variables and their membership maps;
- the production rules;
- the type of logical connectives.

This is a simple task, even in the Legal Field, since the linguistic variables derive immediately from the legal texts under examination. Their membership maps and the production rules can be get from experts knowledge.

Once this has been done, we automatically get, from the tool, the required inferential mechanism.
The general setting so obtained, seems to be of interest in its own right, and perhaps useful for other applications.
In fact, we have, in a sense, proved the feasibility of setting up a (possibly large) system of «legal rules», within a reasonable system to «apply» such rules and even to «interpret» them.
It is important to stress that, at this point, the problems discussed above have not disappeared but require a proper modelling, involving a number of factors (some of which are treated in the next paragraph).
Of course, there are many way to do this - depending also on what aspects are under consideration.
So we consider our rules system as an example, and we think that it is useful if different «computational» experiments will be done by other people. To this aim, we will make available the modified tool as soon as it is completed.
The modifications will include:
- support of «crisp variables»;
- capability of forward chaining;
- learning modules.
It is clear that such modifications are necessary for using the tool in the Legal Field.

1.6. What Remains to Do

In addition to rules and inferential mechanism, many other aspects remain to be modelled, in particular:
- the relationships between legal texts and «rules» therein embedded;
- the cross references, as well as the other relations, between rules;
- the user perception of the legal system he is dealing with.

This is a formidable task, indeed. On the other hand, it is not hopeless that if we start with suitably simplified assumptions, we can get useful approximations to be refined by trials and errors.
For further details see Section 4.

2. Requirements on Models of Legal Rules

In this Section, we illustrate some of the most relevant characteristics of our legal rules. (We do not pretend to be complete, and we refer to [EB],
[GC], [FC], [AC], [RD], [AG], [RG1], [RG2], [HH], [JS], [EZ], [KK] for a proper treatment of this important point).

We have singled out the following ones:
- the _formal_ aspect of the rules, i.e. the fact that we want to use them as formal rules to mechanically compute «consequences», when they are applied to suitable «input cases»;
- the _logical_ aspect of the rules, since we want a semantical meaning attached to our rules;
- the _legal_ nature of the rules, which consists on the fact that they are intended to represent, in some sense, _real legal rules_;

2.1. The Formal and Logical Aspects

We simply refer to usual treatment of such requirements in _theoretical computer science_ fields, as _expert system theory_ or _automatic theorem proving theory_. [WA], [ER], [SD]

2.2. The Legal Aspects

Jurists have pointed out that legal rules have (generally speaking) the following characters:
- external character;
- hypothetical character;
- generality;
- abstractness;
- imperative character;
- impersonal character;
- cogency;

Some of the previous listed characters are of no relevance to our purposes, since they are automatically ignored by the model, because of its formal nature. So, in particular, have no relevance:
- the external character, since we are not interested in mental interior states;
- the cogency aspects, which are not considered in the structure of the rules, whilst they concern the modelling of user-system interaction.

Instead, the other characters _must_ be considered in formal rules. In particular:
- hypothetical character: formal rules must have an hypothetical form _if X then Y_;
- generality and abstractness: formal rules must have a *universal form*, that is there are free variables inside them; *application* of the rules to a specific case involves the substitution of concrete values for variables.

2.3. *The Semantics of Rules: Laws as Control Means*

All legal systems can be considered as a set of norms, which are of the same structure: generally speaking, they can be seen to have an hypothetical form.

Human behaviour becomes juridically relevant because it has been previously evaluated or qualified by norms.

On the other hand, what kind of meaning can we assign to such evaluation process?

From a sociological point of view, laws can be considered as social control means or as social cohabitation means.

So, e.g., penal laws, intended as a control of dangerous behaviours, can be considered as means to avoid or to react (or repair) against some socially hurtful facts or events. With the exception of particular cases, laws do not intend to completely determine the behaviour of individuals, rather they try to compel it into *acceptable ranges*.

In this respect we report the following citation: «We view the legal system as an artificial, man made system, whose task is that of coordinating local actions in order to achieve a satisfactory local reward of life and freedom so that, out of the achievement of these local goals and out of the satisfaction of the coordinating actions, the global goal of life and freedom is also achieved. Since the model is in continuous operation and we allow for disturbances, any changes in society must ultimately effect the legal system, that will have to change to match up the local and global version of the fundamental principles.» [CVC].

Therefore, we expect formal rules to be applied in an environment which is not completely deterministic, therefore permitting soft inferences as, e.g. *probabilistic* or *analogical* ones. For more on this point, we refer to next Section in connection with *fuzzy logic*.

3. *Fuzzy Logic and Legal Rules*

3.1. *Inference Mechanism: Fuzzy Logic*

Nowadays fuzzy logic has a great field of application, from the control systems, to medical expert systems, to financial decision making systems.
Moreover, there is a general agreement about its applicability to the legal field [LP], [TM].

We assume that a legal decision may be considered in some way as a control problem (see «Requirements on models of legal rules»); therefore, we can take advantage from the experience made in using fuzzy logic for such problems.

In the A.I. of the Legal Field, fuzzy logic has two main objectives:
- representation of ambiguity in legal language;
- representation of approximate nature of legal reasoning.
(see [GS2], [TM]).

In a legal decision a lot of factors are involved, on which we obtain a judge's sentence. Such factors need an evaluation in which the judge is unavoidably affected by subjective considerations, coming from his own professional experience. So, to the natural ambiguity of legal language, we have to add the uncertainty introduced by judge's interpretation of the legal text.

Now, we explain how we can apply fuzzy logic in this context.

First of all it's necessary to have a tool to represent a legal rule in a suitable way, maintaining its ambiguity with a certain degree of approximation and without tying it to a specific interpretation. Moreover, it's necessary to represent, in some way, the legal reasoning, saving its ambiguity and approximation.

In a FLC (Fuzzy Logic Controller) the use of fuzzy logic allows the development of systems with the following features [YRP]:
- sophisticated knowledge and rich human experience can be incorporated into the fuzzy knowledge base in an almost natural language;
- the incorporated knowledge is not necessarily precise and complete;
- the input facts to be assessed in fuzzy inference are not necessarily clear-cut nor do they have to match the given knowledge exactly;
- partially matched conclusions can be inferred from the fuzzy facts and the established fuzzy knowledge base;
- fuzzy inference is similar to human reasoning;
- more flexibility is available than in traditional logic.

3.2. System Design

For more details see [MC], [YRP].

3.2.1. Rules Design

In order to obtain a practical example of the development method we are going to explain, we have made a simplified version of the Highway
Code which, nevertheless, contains the main notions about drivers behaviour («speed», «precedence», «overcoming», «signals» etc.).

3.2.1.1. Recognition of Linguistic Variables

The main step in the design is the translation from textual rules to fuzzy rules. In a norm written in legal language we can find ambiguous expressions, the meaning of which is related to the common sense (the so-called linguistic variables).

So we have to single out all relevant linguistic variables and to represent them suitably by fuzzy sets. We have chosen, after subsequent attempts (refining), the fuzzy parameters listed below:
- speed;
- road state;
- visibility;
- speed difference;
- aggravating circumstance;
- extenuating circumstance;
- responsibility.

3.2.1.2. Recognition of Crisp Variables

We have also recognized the following crisp (see below) concepts, that don't need any evaluation:
- context (urban centre, speed-way, suburbs-roads, high speedy roads) to fix the speed limits;
- place (cross-roads, curve, tunnel, traffic-lights, school or hospital nearby etc...);
- precedence;
- gear (over-cross, turning to right, turning to left, etc.);
- special cases (carriage of patients, public assistance cars, etc.);
- kind of judge (strict, normal, tolerant);
- signals (stop, over-cross forbidden, stop, give precedence, etc.);
- car-state (tires-wear, breaks-wear, headlights out of use, etc.);
- driver-state (drunkness, without glasses, sleepiness, inattention, etc.).

Recall that in the fuzzy logic terminology, we call crisp parameters that always have a completely determined value.

3.2.1.3. Consideration on FLC

Before going on it's convenient to remember some considerations on FLC design.

In designing an FLC the principal factors to be kept in mind include the following ones (see: e.g. [TAS], [DHR]):
- The actual inputs and outputs and their universes of discourse, i.e. the range of values which each may take.
- The scale factors of the input-output variables.
- The fuzzy membership functions to be used in setting the fuzzy values for each input and output variable.
- The fuzzy control rule base.

Although the choice of the number, range and shape of membership functions for a variable is ultimately based on subjective design choices and evaluation of the resulting system performance, the following guidelines are useful:
1. Symmetrically distribute the fuzzy sets across the defined universe of discourse.
2. Use an odd number of fuzzy sets for each variable – this ensure that some fuzzy set will be in the middle. Five or seven fuzzy sets are fairly typical.
3. Overlap adjacent fuzzy sets to ensure that no crisp value fails to correspond to any set, and to help ensure that more than one rule is involved in determining the output.
4. Use triangular or trapezoidal membership functions as these require less computation time than other types.

3.2.1.4. Setting the Membership Functions and Fuzzy Rules

The main aspects of the representation of the fuzzy variables are membership functions and, respectively, the rules structure.

In fact the setting of fuzzy sets, that determines the allowable values for variables, has a great influence on the system performance.

Typically fuzzy sets are defined by linguistic expressions such as «tall», «short», «hot», «very hot» etc., to keep a close distance with natural language. The number of fuzzy sets defined for a variable, determines the allowable values for it, and the precision or accuracy of the control we want to obtain. Every fuzzy set is defined or numerically as a list of couples <argument,degree> (where argument is one of the allowable value of the variable, and degree is any number between 0 and 1), or mathematically as a membership function. The most used functions are: triangular, bell-shaped, trapezoidal, exponential, S-function.

Ex: (Definition of the variable «visibility» in the fuzzy language made by Dublin City University FLC)
INPUT visibility TYPE short
  RANGE 2 to 200 STEP 2
  FUZZY bad FUNCTION triangle (20 42 64)
  FUZZY very bad FUNCTION esseneg (5 15 25)
  FUZZY normal FUNCTION trapezium (61 101 105 145)
  FUZZY good FUNCTION triangle (100 125 150)
  FUZZY very good MEMBERS (110,0.5) (140,0.7) (160,1) (170,1)

Rule 1 IF visibility IS bad AND roadstate IS medium THEN speed = slow
Rule 2 IF visibility IS very bad THEN speed = very slow
Rule 3 IF roadstate IS very bad THEN speed = very slow

The starting rule was of this kind:
  The driver must respect the following speed limits:
    50 km/h in an urban centre.
    70 km/h in suburbs-roads.
  ...

These limits are acceptable in perfect environment conditions (visibility, road state, climatic conditions...); the driver must reduce his speed according to the above conditions, to have a safe driving.

(The definitions of road state and speed have been omitted because they were similar to that of visibility).

3.2.2. System Structure

After having established which are the fuzzy variables, we have to distinguish them in input variables and output variables. An input variable may appear only in the antecedent of a rule and an output one only in the consequent of a rule. This causes that the system must be structured in order to allow a forward inference, because the value of an output variable, result of a fuzzy inference, can’t be used as input in a rule for a subsequent inference. So, to permit forward inferences, we have to organize a system stratification with different levels.

There are two knowledge bases (one for the determination of the speed, and the other one for the determination of the responsibility), and a C module (for the evaluation of the aggravating and extenuating factor).

The result of law application is calculated on the base of the environment’s state, that is road state, visibility, etc.
3.2.2.1. First System Level

First level knowledge base rules are like this:

CONDITION context
   IF visibility IS '...' AND road state IS '...' THEN speed = '...'

where '...' indicates a fuzzy variable set, and «context» is a crisp variable of the kind explained above (see paragraph 2.1.2).

Rules can be use boolean conjunctions e disjunctions

CONDITION context
   IF (visibility IS '...' OR visibility IS '...') AND road-state IS '...' THEN speed = '...'

Because the DCU compiler cannot use boolean variable, the system had to use a C module to manipulate crisp value as over-coming, precedence, etc.

In fact, between the two levels there is a C module that computes the speed difference, aggravating and extenuating factors that cause an increase, resp. a decrease of responsibility.

Aggravating factors enter into play, when the driver does not follow what is prescribed by the law: when he drives being not in perfect conditions (e.g. without glasses), when his vehicle is not in perfect conditions (e.g. brake wear).

Ex:

IF driver_state IS 'drunkenness' THEN aggravating _sfc=max
IF car_state IS 'tyres-wear' THEN aggravating _sa=max

Aggravating factors are then put into an aggravating fuzzy variable with range [1, 10].

Ex: aggravating definition

INPUT aggravating TYPE short
   FUZZY min FUNCTION esseneg ()
   FUZZY med FUNCTION trapezium ()
   FUZZY max FUNCTION esse ()

Extenuating factors take into account the situations that do not depend on the driver (e.g. momentary sickness), or special situations (e.g. transport of sick person).

As the aggravating factors, all extenuating ones are put into an extenuating fuzzy parameter, defined as the previous one.
3.2.2.3. Second System Level

In the second level of the system there are rules for the responsibility evaluation, based on the speed difference (the difference between the driver speed and the speed admitted by the law), and the aggravating and extenuating factors. As in the «real world», the responsibility also depends on the type of judge (see Paragraph 2.1.2).

To fuzzy variables «difference», «aggravating» and «extenuating» are assigned 3 fuzzy set: min, med, max.

The type of judge increases or decreases the driver responsibility; such types, however, cause not too different judgements. They can be seen as parameters that give a lower and an upper bound to the responsibility: the difference between the permissive judgement and the severe one, is less than 10%.

The fuzzy set responsibility has 5 fuzzy sets and a [0, 100] range.

```
OUTPUT responsibility TYPE short
RANGE 0 TO 100 STEP 1
FUZZY none  MEMBERS (0, 1) (1, 1)
FUZZY very low FUNCTION esseneg (5 10 15)
FUZZY low FUNCTION pigreco (25 26)
FUZZY med FUNCTION pigreco (48 36)
FUZZY high FUNCTION pigreco (75 26)
FUZZY very high FUNCTION esse (85 90 95)
FUZZY total  MEMBERS (99, 1) (100,1)
```

Second knowledge base rules are of the type:
```
CONDITION judge type
IF difference is '...' AND aggravating is '...' AND extenuating is '...' THEN responsibility = '...'
```

Rules take into account all parameter configurations of «difference», «aggravating» and «extenuating».

The judgetype parameter changes the responsibility using a fuzzy HEDGE operator (see below).

3.2.3. Hedge Operator

This type of operator changes the form of the fuzzy set of the variable to which is applied.
Ex:  
**CONDITION** judge type = severe  
IF difference IS max AND aggravating IS medium THEN responsibility = very high  
where «very» is an hedge operator defined by:  
**HEDGE** very X ^0.33  
following the «powered» approach that increases the fitness value of the high fitness value elements of the fuzzy set.  
Following the «shifted» approach the fitness value element are shifted forward or backward.

Ex
**HEDGE** plus X+20  
**HEDGE** minus X-20  

Which type of modifier has to be used depends on the meaning of the modified fuzzy set.

To obtain an environment sensitive speed we can use 2 ways:  
- we can define the fuzzy set for all the type of environment.  
- we can define one type of fuzzy set for the speed and then we can modify them with an hedge operator.

Ex: fuzzy sets for the speed in built-up area (50 km/h limits)  
**FUZZY MB FUNCTION** ()  
**FUZZY B FUNCTION** ()  
**FUZZY M FUNCTION** ()  
**FUZZY A FUNCTION** ()  
**FUZZY MA FUNCTION** ()

Fuzzy set for the speed outside a built-up area (70 km/h limits) using an hedge operator:  
**HEDGE** plus X+20  

IF visibility IS ‘...’ AND road-state IS ‘...’ THEN speed = very ‘...’  

In this case the use of powered hedge operator would not give the required result.

3.3. *Examples on System Behaviour*  
**Example 1**  
context: urban center  
special events: none
gear: overcross
place: cross-roads
signals: overcross denied
car-state: none
driver-state: without glasses
road-state: medium
visibility: 30m
speed: 90 km/h
precedence: to periphery street
legal speed: 50 km/h
difference between real speed and legal speed: 40km/h
aggravating circumstances: about overcross and driver-state
attenuating circumstances: none
responsibility determined by a normal judge: 100%
responsibility determined by a severe judge: 100%
responsibility determined by a permissive judge: 100%

Example 2
context: urban center
special cases: none
gear: normal
place: pedestrian crossing
signals: none
car-state: none
driver-state: none
road-state: very good
visibility: very good
speed: 30 km/h
precedence: to pedestrian
legal speed: 50
difference between real speed and legal speed: none
aggravating circumstances: none
attenuating circumstances: none
responsibility determined by a normal judge: 0%
responsibility determined by a severe judge: 0%
responsibility determined by a permissive judge: 0%

Example 3
context: urban center
special cases: patient carriage
Intrigila, Cianci, Damiani, Di Salvatore, Centanni / Computational Models

gear: normal
place: near schools or hospitals
signals: give precedence
car-state: none
driver-state: none
road-state: normal
visibility: normal
speed: 81 km/h
precedence: none
legal speed: 50 km/h
difference between real speed and legal speed: 31 km/h
aggravating circumstances: about precedence
attenuating circumstances: about special cases
responsibility determined by a normal judge: 89%
responsibility determined by a severe judge: 93%
responsibility determined by a permissive judge: 84%

Note: although there is an attenuating circumstance concerning «special cases», missing precedence and high speed caused high responsibility.

Example 4

context: urban center
special cases: none
gear: normal
place: cross-roads urban-peripherycal
signals: give precedence
car-state: tires wear
driver-state: none
road-state: normal
visibility: good
speed: 68 km/h
precedence: to peripherycal street
legal speed: 50 km/h
difference between real speed and legal speed: 18 km/h
aggravating circumstances: about car-state
attenuating circumstances: none
responsibility determined by a normal judge: 55%
responsibility determined by a severe judge: 59%
responsibility determined by a permissive judge: 52%
Example 5
context: suburbs-roads
special cases: none
gear: normal
place: cross-roads urban-peripherycal
signals: give precedence
car-state: tires wear
driver-state: sleepness
road-state: bad
visibility: bad
speed: 67 km/h
precedence: none
legal speed: 51 km/h
difference between real speed and legal speed: 16 km/h
aggravating circumstances: about car-state and driver-state
attenuating circumstances: none
responsibility determined by a normal judge: 100%
responsibility determined by a severe judge: 100%
responsibility determined by a permissive judge: 100%

Example 6
context: suburbs-roads
special cases: tire burst
gear: in bend
place: none
signals: continuous line
car-state: tires wear
driver-state: none
road-state: normal
visibility: normal
speed: 53 km/h
precedence: none
legal speed: 60 km/h
difference between real speed and legal speed: none
aggravating circumstances: about car-state
attenuating circumstances: about special cases
responsibility determined by a normal judge: 88%
responsibility determined by a severe judge: 96%
responsibility determined by a permissive judge: 84%
Note: although there is an external factor (tire burst), responsibility is equally high because of the relevance of tires wear.

Example 7

context: motorway
special cases: none
gear: overcross
place: none
signals: none
car-state: none
driver-state: none
road-state: normal
visibility: good
speed: 142 km/h
precedence: none
legal speed: 130 km/h
difference between real speed and legal speed: 12 km/h
aggravating circumstances: none
attenuating circumstances: none
responsibility determined by a normal judge: 10%
responsibility determined by a severe judge: 12%
responsibility determined by a permissive judge: 7%

Example 8

context: motorway
special cases: none
gear: turning to left
place: direction lane
signals: give precedence
car-state: none
driver-state: inattention
road-state: good
visibility: very good
speed: 69 km/h
precedence: none
legal speed: 50 km/h
difference between real speed and legal speed: 19 km/h
aggravating circumstances: about precedence and driver-state
attenuating circumstances: none
responsibility determined by a normal judge: 100%
responsibility determined by a severe judge: 100%
responsibility determined by a permissive judge: 100%

Example 9
context: high speed-roads
special cases: none
gear: normal
place: none
signals: none
car-state: broken light
driver-state: none
road-state: normal
visibility: normal
speed: 114 km/h
precedence: none
legal speed: 90 km/h
difference between real speed and legal speed: 24 km/h
aggravating circumstances: about car-state
attenuating circumstances: none
responsibility determined by a normal judge: 56%
responsibility determined by a severe judge: 58%
responsibility determined by a permissive judge: 53%

Example 10
context: high speed-roads
special cases: patient carriage
gear: normal
place: none
signals: none
car-state: broken light
driver-state: none
road-state: normal
visibility: normal
speed: 114 km/h
precedence: none
legal speed: 90 km/h
difference between real speed and legal speed: 24 km/h
aggravating circumstances: about car-state
attenuating circumstances: about special cases
responsibility determined by a normal judge: 52%
responsibility determined by a severe judge: 56%
responsibility determined by a permissive judge: 50%

Note: we remark that this example compared to the previous one, points out that the attenuating circumstances evaluated as important, decreased the responsibility.

4. Further Work on Modelling and Computational Tool

4.1. Some Considerations on Text Generation

This section is dedicated to the creation of a natural language text representing fuzzy rules. Text created should be similar to text representing real norm, so it should be vague, with ambiguity and with many cross-references.

To obtain these results the simplest way is to use a set of typical norm textual forms that will be used to express our fuzzy rules.

Before doing this it is necessary to make some operations on the rules to be treated, such as normalization, groupation, and so on.

With the normalization process we want to give to fuzzy rules a standard form, to make easier their translation.

Rules that have two or more variables in the consequent are replaced by rules with the same antecedent and only one variable in the consequent;

Rules containing OR in the antecedent are replaced by two rules containing the OR arguments as antecedent.

Eg:
IF visibility is bad THEN speed is low ALSO prudence is high
becomes
IF visibility is bad THEN speed is low
IF visibility is bad THEN prudence is high

IF visibility is bad AND (road state is good OR road state is very good) THEN speed is low
becomes
IF visibility is bad AND road state is good THEN speed is low
IF visibility is bad AND road state is very good THEN speed is low

The groupation process is used to group fuzzy rules that express the same norm. In fact for a sharp control, a single concept, when expressed
in fuzzy logic (fuzzy variable), is divided in some cases: eg the fuzzy variable visibility has been divided into 5 fuzzy sets: very bad, bad, normal, good, very good.

The same division is made for rules, so a norm like

**speed must be proportional to visibility**

in our case could be translated into 5 rules, taking into account the different definition of visibility

- **If visibility IS very bad THEN speed = very low**
- **If visibility IS bad THEN speed = low**
- **If visibility IS normal THEN speed = normal**

... and so on. We want to do the opposite operation.

Once having grouped rules representing the same norm, we can translate them into natural language, using the text «template», alone or combining two or more of them.

To chose the right template to be used, we should have an idea of what the fuzzy rule expresses. Our idea is to analyze the influence that input variables have on the output variable, in the groups of fuzzy rules obtained through the group operation. The method proposed is very simple and it is borrowed from mathematical multi-dimensional derivation.

Rules contained in these groups have the same input and output variables. Chosen a variable, we look for all rules having the same context. Having the same context we are sure that the output variable values in these rules is influenced only by the chosen variable values, so we can try to interpret the existing relationship. In this respect we have to know if a fuzzy set indicates a higher or lower range of values compared to another one of the same linguistic variable. In this way, we want to sort the fuzzy sets of the variable. We can do this directly with a thesaurus or looking to the definition of fuzzy sets and hedge functions.

Besides, the validity of our deduction is tied to the number of logical connectives in the antecedent of the rules: the validity is as high as the number of conjunctions is low.

Eg:

- **If visibility IS bad AND road state IS good THEN speed IS very low**
- **If visibility IS bad AND road state IS very good THEN speed IS low**

In these two rules the variable road state has the same context (visibility has the same value). Looking at the values of road state and speed we can
deduce that road state and speed are proportional. From the thesaurus we get these information:
- for road state good is less than very good
- for speed very low is less than low.

Now is it possible to translate fuzzy rules into text.

To obtain more vagueness and ambiguity we could use a thesaurus to interchange words in the texts with word of the same (or similar) meaning in the thesaurus. To have an effect of ambiguity, the word that will substitute the concept expressed, should be selected randomly, from a thesaurus, suitably built up.

4.2. The Tool Modifications

4.2.1. The Possibility to Deal with Boolean Variables

Many of the rules used in the system must be applied only if certain conditions are true. This conditions represent facts, so their values are boolean i.e. crisp values and the original tool’s rules could not deal directly with such type of information. Our modification has been introduced to solve this problem.

We added a new type of variable (an input variable): the crisp variables. These are boolean variables, and can be used in the head of fuzzy rules, with the token 

Eg:

a fuzzy rule that does not use boolean variable
\[
\text{IF visibility is normal THEN speed} = \ldots
\]
a fuzzy rule that uses boolean variable
\[
\text{CONDITION context} \\
\text{IF visibility is normal THEN speed} = \ldots
\]

The second rule indicates a fuzzy rule (IF visibility is normal THEN speed = ...) that can be applied only if the boolean variable context is true.

When the tool generates the C language program corresponding to the fuzzy program, rules that use boolean variables, are translated in a different way: the code corresponding to the fuzzy rule is closed into an if statement. The if statement controls the value of the crisp variable, so the code corresponding to the fuzzy rule is executed only if this value is true.
Eg:

```java
CONDITION context
    if visibility is normal THEN speed = ...
```

is translated in

```java
if (context) {
    // code of the fuzzy rule
}
```

### 4.2.2. Forward Chaining

To obtain the forward chaining we divide the fuzzy program into subprograms.

1. Each subprogram computes the value of one (or more) fuzzy variable and it is translated in C language as the original tool does.

2. When all the subprograms have been translated, a final procedure is added to the C program. This procedure calls in sequence all the routines corresponding to each subprogram.

3. At the beginning of the program there is the declaration of the variables used in the fuzzy program. Each sub program then declares the variables that it uses and if it is uses as input or not.

As an example if we want to calculate the variable $a$ using the variable $b$, and we want to calculate the variable $c$ from the variable $a$, we have to define 2 subprograms: the first that computes $a$ from $b$ ($b$ as input, $a$ as output), and the second that computes $c$ from $a$ ($a$ as input, $c$ as output). The final procedure will call the two procedures in sequence.

So we have obtained forward chaining by only adding a new software layer, and the original tool structure has not been changed.

```plaintext
PROGRAM name of the program
variable definitions
    SUB
    definition of the I/O variable
fuzzy subprogram
    END

    SUB
    definition of the I/O variable
fuzzy subprogram
    END

... 

REASONING type of reasoning
```
4.2.3. Learning Module

Our use of fuzzy logic can be divided into two levels (as the DCU language does):

- the fuzzy set definition level;
- the rule definitions level.

The fuzzy set definition level is a low level program: here we find the variable and hedge function definitions. These are concepts that will be used later in the program.

The rule definitions level consists of the high level program: here are located the instructions that determine the output values obtained by the input values. In this part we find only linguistic variables, hedge operators etc., and nothing that can deal with vagueness, since this part is included only in the fuzzy set definitions.

According to [YRP], [TAS], the fuzzy set definition phase is a very important one, in building a fuzzy application, and its tuning is a very difficult task.

Since a fuzzy rule can be easily represented by a neural network [YRP], [TAS] (we are talking about a slight modification of the rules used in DCU fuzzy tool), we could use the error back-propagation algorithm to adjust weights in order to obtain the desired tuning.

The rules of the DCU tool have the following structure

\[
\text{IF } a \text{ is } b \text{ THEN } c = d
\]

The value of \( c \) is a function of the fuzzy value of the \((a \text{ is } b)\) statement, where \( b \) is a fuzzy set.

\((a \text{ is } b)\) returns the value of \( a \), with respect to the fuzzy set \( b \).

Using some weights we could change dynamically a fuzzy set definition in a rule shifting or stretching it, say by an hedge operator, influencing the final result.

This type of operation can be used also in the defuzzification operation.

Schematically, we have:

\[
\text{INPUT } \Rightarrow \text{ WEIGHTED(⊗) FUZZY SETS } \Rightarrow \text{ RULES } \Rightarrow \\
\Rightarrow \text{ WEIGHTED(⊗) FUZZY OUTPUT } \Rightarrow \text{ OUTPUT}
\]

(⊗) indicates where learning can be applied.
References

Fuzzy Logic and Control


Law and Legal Rules


Artificial Intelligence and Related Topics


Information Retrieval

Artificial Intelligence and Law

[CB] C. Beardon, Prolog, Logic and Legal Rules.
[CVC] Francisco A. Casiello, Juan M. Villaruel, Carlos M. Casiello, Hierarchical Multilevel Modelling of Legal Systems and of Their Impact on Society, in [BB].