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

J.C. Hage, Monological reason based reasoning: 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. 77-91, 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

MONOLOGICAL REASON BASED REASONING

Jaap Hage
University of Limburg
Department of Metajuridica
P.O. Box 616
6200 MD Maastricht
The Netherlands

Summary

This paper presents a theory of Reason Based Reasoning (RBR). RBR comes in a monological version, in which it is a theory of rational belief. It also has a dialogical version, which makes it a theory of argumentation. In both versions, the notion of a reason to belief takes a central place. In this paper the monological version of RBR is exposed, illustrated with an example from the legal domain, and formalized.

1. Introduction

There are reasons to doubt the satisfactoriness of classical logics (first order predicate logic, standard deontic logics) for the analysis and evaluation of legal reasoning [Hage 87b and 91]. In this paper I propose an alternative for these logics by means of a formalized theory about Reason Based Reasoning (RBR).

RBR comes in two versions, a monological and a dialogical one. The monological version amounts to a theory about rational belief (revision). It indicates which conclusions should be accepted, and which ones rejected on basis of a number of accepted beliefs and rules of inference. As such it exhibits similarities with truth maintenance systems [cf. Doyle 79 and De Kleer 86]. Because of lack of space, these similarities and the differences with truth maintenance systems are not elaborated in this paper.

The dialogical version of RBR offers additional rules for dialogues, by means of which can be determined which party in a dialogue wins a discussion. This version takes more peculiarities of legal reasoning into account, especially the division of the burden of proof. The dialogical approach builds upon the work of e.g. [Lorenzen and Lorenz 78].

In this paper the monological version of RBR is presented, with a focus on its application to legal reasoning.

2. The main ideas behind RBR

2.1. Differences with classical logics

A first main difference between RBR and classical logics is that in RBR arguments are based on reasons adduced for or against a conclusion. The presence of a reason for a conclusion does not guarantee acceptance of the conclusion. There can be reasons for and against the same conclusion. Whether the conclusion ought to be accepted depends ultimately on the weighing of reasons.

A second major difference is that RBR models a process, while classical logics model the contents of an amount of information. This feature of classical logics is reflected in the presence of semantics and the close relationship between the syntactic and semantic characteristics of these logics (stress on completeness and consistency). RBR lacks semantics and is far less concerned about consistency and circularity than classical logics.

The process which RBR attempts to model is human belief revision. But the endeavour is not to model actual belief revision, but rational belief revision. Since the nature of rationality is far from clear, there are several ways to elaborate RBR. At the end of this paper some of these ways are shortly hinted at. They are concerned with the question how much effort must be spent on keeping the available information consistent.

The question after the effort on consistency is not only important with respect to modeling human belief revision, but also with respect to the possibility of machine implementation of RBR.

A third difference with classical logics is that RBR uses two basic categories, namely propositions and rules of inference (rules), where classical logics only use propositions. Indeed, classical logics also know rules of inference, but they use them in quite a different fashion than RBR does. What are rules in RBR, would be universally quantified material implications in classical logics. Rules differ from these implications amongst others in that they cannot be used for Modus Tollens-like inferences and in that they have scope conditions.

Propositions in RBR are belief contents. They are comparable to Frege's judgment contents [Geach and Black 1952, p. 2]. Propositions can be accepted or not. It is possible that neither a proposition, nor its negation is accepted. Therefore it is not possible to derive a proposition from the fact that its negation is not accepted, or the other way round. In this respect RBR is related to three-valued logics [cf. Haack 76, ch. 11].

RBR is non-monotonic. It is possible to retract information from the data RBR is working upon (the belief set). To achieve belief maintenance, in RBR belief elements are accompanied by their justifications. If a justification becomes invalid, the belief element is retracted.

These justifications have not only a function in the context of belief retraction. They also serve to relativize beliefs with respect to their origins. Propositions and rules are accepted relative to their justifications. There is no such thing as absolute acceptance. This feature of RBR, which cannot be discussed elaborately in this paper, is important for e.g. the handling of conflicting prima facie deontic judgments [Ross 30; Searle 78].

For more details concerning the philosophical background of RBR, the reader is referred to [Hage 87a and 91].

2.2. Standard rules and reasons

According to RBR, rules make propositions into reasons. By adducing a proposition as a reason, one presupposes the acceptance of the rule which makes this proposition into a reason. By accepting the rule, one accepts certain facts as a reason to belief or not belief a proposition, or as a reason to accept or not accept some other rule.

Rules not only have conditions, but also a scope. The use of rules presupposes that certain conditions are fulfilled. For instance, the use of a rule of Dutch Penal Law presupposes that the judged act was committed in the Netherlands, or that the actor has the Dutch nationality [Dutch Penal Law Code, sections 2-5].

The conditions of scope must be distinguished from the standard conditions of the rule. In law, the burden of proof that the scope conditions are not satisfied usually rests on the party which wants the rule not to be applied. In the monological version of RBR, the scope conditions of rules are satisfied by default. They are represented by means of exclusionary reasons.

2.3. Meta-rules and meta-reasons; a legal example

A standard reason is a proposition used in an argument to support a conclusion. A meta-reason is a reason which influences the functioning of a rule. For instance a meta-reason can influence the weight a rule assigns to a reason. Or a meta-reason cancels the application of a rule in a concrete case, since it holds that the case is outside the scope of the rule. Meta-reasons excluding the applicability of rules are called exclusionary reasons.

A proposition can be a reason and a meta-reason simultaneously. Take the following example: A driver approaches crossroads. The traffic lights are red, implying that he should stop. However, a police officer, regulating the traffic, signals him to drive on. What should he do? In Dutch traffic law there is a conflict rule stating that directives from police officers precede general rules of traffic. So the driver should obey the police officer, and continue his way.

How is this to be construed? The signal of the officer is a reason to drive on. This is the reason on which the driver should base his belief about what to do (and his behaviour too). But what happens with the reason that the traffic light was red? This reason has lost its force, because of the signal of the police officer. That the officer signalled the driver to continue is both a reason why the driver should continue, and an exclusionary reason cancelling the force of the red traffic light. As a consequence, the fact that the traffic light is red is not a reason to stop anymore.

In general, legal conflict rules make propositions which are standard reasons on basis of one of the conflicting rules, exclusionary reasons with regard to the other rule. Conflict rules can as a consequence be considered meta-rules, influencing the relevancy of other rules. Meta-rules underlie meta-reasons.

2.4. Weighing reasons and hard cases

A basic idea behind RBR is that a specific conclusion can be drawn as a result of weighing the reasons for and against that conclusion. But very often it turns out that only reasons for or only reasons against a conclusion are available. All reasons pleading in the other direction are excluded. Frequently, a reason which superficially seems to be a reason against a conclusion, actually is a meta-reason defeating a reason for that conclusion.

If all reasons against a conclusion are defeated and only reasons pro remain, no weighing of reasons is necessary anymore. In my opinion, this situation or its reverse, where only reasons against remain, occurs more often in actual reasoning than the situation where reasons have to weighed. The basic idea behind RBR is correct but often irrelevant. The part about meta-reasons is actually more important.

It is a happy situation that the role of weighing reasons is small, because the idea of weighing reasons suffers from serious problems. It is not clear what weighing reasons amounts to, nor what the weight of a reason exactly is. We all are familiar with the situation where there are reasons for and against a conclusion and where we must somehow make up our minds. We use the expression 'weighing reasons' to describe the process of coming to a decision. But we are seldom or never aware of something like the weight of a reason, let alone of a process of actually weighing reasons. Weighing reasons is a metaphor for a phenomenon we know very little about, and which is presumably better analyzed in causal than in reason related terms.

If we have reasons for and against a conclusion, none of which can be excluded, we are stuck with a problem. No rational procedure exists to solve it. Admittedly, human beings are most of the times capable of handling this problem. But they do not use criteria to solve it and they cannot justify the outcome otherwise than by saying that the decision was the outcome of the metaphorical weighing of reasons. When, in legal theory, we speak of hard cases, I think we almost always have in mind cases, with undefeated reasons for solutions pointing in different directions. Hard cases are hard, because we are left without a rational procedure to solve them.

In RBR a very simple mechanism handles the weighing of reasons. The weights of the reasons pro and contra a thesis are summed and the highest weight wins. Probably this solution will not be adequate for many situations. The introduction of meta-reasons influencing the default weight of reasons might solve a number of problems, but this topic is too complex to explore in the remainder of this paper. For practical purposes, the use of exclusionary reasons is more important.

3. The formalization of RBR

3.1. The Belief Set and belief elements

The monological version of RBR uses as a starting point for the analysis of reasoning a system (a person or a machine) disposing of a set of beliefs. This belief set includes zero or more elementary propositions and/or rules. To avoid redundancy and computational overhead, compound propositions are not included. These elementary propositions and rules are said to be in the belief set, to be elements of the belief set, or to be belief elements. I will use the capital letter B to refer to the belief set. The manipulation of the contents of B is what RBR is all about.

The capital letter E is used as a constant for belief elements. As variable for belief elements, the small letter e, sometimes followed by a subscript will be used. Accordingly, e, e_1, e_2 … e_n are variables for belief elements.

There are no formal restrictions on the contents of B, other than those that follow from the rules for adding belief elements to B and retracting them from B (rules

R2a and R2b, to be discussed later).

3.2. Propositions

Elementary propositions

Propositions can be elementary or compound. Elementary propositions are negated or not.

In the following, the capital letter P, sometimes followed by a subscript, will be used for propositional constants. Accordingly, P, P_1, P_2 … P_n are propositional constants, which can be used for both elementary and compound propositions. Non-negated elementary propositional constants can be preceded by the negation sign. The resulting negated propositional constants ~P, ~P_1 … ~P_n always stand for negated elementary propositions.

Propositional variables

For propositional variables, the small letter p, sometimes followed by a subscript will be used. Accordingly, p, p_1, p_2 … p_n are propositional variables. There are also negated propositional variables: ~p, ~p_1 ... ~p_n.

Non-negated propositional variables can instantiated by both negated and non-negated elementary propositions, and by compound propositions. A negated propositional variable can only be instantiated by a negated elementary proposition.

Lists and sets of propositions

A list of propositions consists of one or more propositions, separated by comma's, and surrounded by brackets, e.g. [P_1, P_4, ~P_7]. A list of propositions is the representation of the corresponding set of propositions.

In the following, the capital letter L, sometimes followed by a subscript, will be used for proposition list constants. As variables for proposition lists the small letter l, sometimes followed by a subscript, is used.

Compound propositions

Lists of propositions can be transformed into a compound proposition by means of the conjunction or the disjunction function.

A compound proposition constructed by means of the unary conjunction function and() is called a conjunction. The elements of the list are called conjuncts. A conjunct must either be an elementary proposition, or a disjunction.

A compound proposition constructed by means of the unary disjunction function or() is called a disjunction. The elements of the list are called disjuncts. A disjunct must either be an elementary proposition, or a conjunction.

Examples of compound propositions are:

and([P_1, P_4, P_7])
or([P_2, ~P_{13}, P_7])
and([P_2, or([P_4,P_6])])
or([and([P_{55}, P_6]), P_4, and([~P_7, P_8])]).

Where in the remainder of this paper the word 'proposition' is used without further specification, both elementary and compound propositions are meant, unless the contrary is clear from the context.

3.3.Rules

Standard rules

Standard rules are constructed by the four-argument function rule(). The first argument of this function is the rule identifier, a positive integer. The second argument consists of a proposition, constituting the conditions of the rule. The third argument is a non-negated elementary proposition or a rule, constituting the rule conclusion. And finally, the fourth argument is a non-zero integer, indicating the default weight of the reasons constituted by the rule.

Examples of standard rules are:

rule(23, P_2, P_4, 4)
rule(12, and([P_2, or([P_5, P_6])]), rule(100), 3)
rule(5, or([P_4, P_6]), P_{45}, -3).

Meta-rules

Meta-rules excluding the application of other rules differ from standard rules in that the conclusion part is replaced by excludes(i), where i is the identifier of the excluded rule. A second difference is that the fourth argument, concerning the weight of the reasons is absent. For instance, rule(2, P_{12}, excludes(8)). This second difference is motivated by the assumption that there is no need to weigh meta-reasons (which are based on meta-rules).

Such meta-rules can be used to express a rule's scope conditions. For instance, the example rule can be used to indicate the scope conditions of rule(8).

Sometimes it is more convenient to represent a rule only by means of its identifier, for instance rule(100), as happened in the second example of a standard rule.

3.4. Reasons

Standard reasons

All reasons are based upon rules, of which the conditions are satisfied. The rule on which is a reason is based is said to underlie the reason. The propositions satisfying the rule conditions are said to constitute the reason.

Standard reasons are represented by means of a four-argument function reason(). The first argument is the identifier of the reason, which is identical to and derived from the identifier of the rule underlying the reason. The second argument consists of an elementary proposition or a list of propositions, which counts as the reason. This is the justification by which the rule is satisfied. (The notions of a justification and of rule-satisfaction are discussed in the sections 4 and 5.) The third argument consists of a non-negated elementary proposition or a rule, constituting the conclusion of the reason. And the fourth argument is a non-zero integer, indicating the weight of the reason. Both the third and the fourth argument are identical to respectively the third and fourth argument of the rule underlying the reason.

Examples of standard reasons are:

reason(23, P_2, P_4, 4)
reason(12, [P_2, P_6], rule(100), 3)
reason(5, P_4, P_{45}, -3).

As will be clear, these reasons are based on the rules from the rule-example. In case of reason(12), the second argument differs from the second argument of rule(12) because only P_6 from the alternative conditions P_5 and P_6 is adopted. The same counts for reason(5), where only P_4 from the alternatives P_4 and P_5 of rule(5) was adopted. This will become clear as soon as justifications are discussed.

Sometimes it is more convenient to represent a reason only by means of its identifier, for instance reason(4).

Exclusionary reasons
The representation of exclusionary reasons deviates from that of standard reasons in that the third argument consists of the function excludes() applied tot the identifier of the excluded rule. The fourth argument, concerning the weight of the reasons is absent. An example of an exclusionary reason is reason(4, ~P_{14}, excludes(6)).

Lists of reasons
A list of reasons consists of one or more reasons, separated by comma's, and surrounded by brackets, e.g. [reason(1), reason(4), reason(7)]. In the following, the capital letters RL or the small letters rl, sometimes followed by a subscript, will be used for reason list constants, respectively variables.

Reason list constants and variables can be connected to the propositions and/or rules derived from them. For that purpose they are represented as follows: RL(P_1) is a constant, standing for the list of reasons from which P_1 was derived. rl(rule(i)) is a variable, standing for the list of reasons from which rule(i) was derived.

3.5.The justification of belief elements

Justifications
All belief elements have a justification. This justification is by means of a slash (/) postfixed to the belief element it justifies.

Default elements
In case of elements which are by default in B, the justification is the word 'default'. Default elements of the belief set do not have a proper justification. The word 'default' serves to indicate this. An example of the use of this justification is P_3/ default.

One can think of default elements as being in B without any reason. People sometimes (or maybe even often) have unmotivated beliefs and use unmotivated rules. Default elements are the RBR counterpart of these beliefs and rules. Default elements which are in the belief set have been there from the beginning, since it is not possible to add default elements to the belief set.

Default elements should not be identified with defaults in the sense of conclusions based upon potentially incomplete evidence. All elements of B which are based upon evidence, are based upon potential incomplete evidence. A notion like complete evidence does not exist in RBR.

Inferred elements

Inferred elements belong to the belief set as conclusions from other belief elements. When a belief element is derived, it is added to the belief set with its justification postfixed to it. The justification of an inferred element consists of the list of reasons on which it is based. An example of the use of this justification is rule(5)/RL(rule(5)).

External elements

Not all belief elements stem from other belief elements. Sometimes beliefs are adopted because of other reasons. These other reasons are called external reasons.

External reasons justify elements of the belief system which 'come from outside'. Coming from outside is a metaphor. In the case of human beings, knowledge stemming from sensory perception may be said to come from outside. In the context of a knowledge based system, coming from the outside may mean that rules or propositions are in the belief set because they are added by the user of the system.

External elements are in the belief set solely for external reasons, which are themselves not elements of the belief set. The justification of external elements consists of an integer, representing the weight of the external reasons for which the element is in the belief set. This weight indicator can be used for weighing reasons in case there are internal reasons against the external element. An example of the use of this justification is: rule(25)/4.

External belief elements are mentioned here for the purpose of completeness. Their role within RBR is only treated cursorily in this paper.

Derivable elements

Not all rules and propositions which can be derived from the belief set are belief elements. Only those elements which have actually been derived are in the belief set (next to default and external elements). To define the notion of a reason, we will make use of the notion of derivable belief elements.

Intuitively a derivable belief element either is already in the belief set, or can be derived from it in one or more steps, where the only changes in the belief set during the process of derivation are consequences of this process. Especially changes in the belief set caused by adding or retracting belief elements from outside are not allowed.

D1: Derivable belief elements are the rules and elementary propositions which either actually are in the belief set, or can be derived from the belief set without external intervention.

The validity of justifications

Every element which is added to the belief set is accompanied by a justification. This justification remains unchanged as long as the element remains in the belief set. However, a justification may become invalid while the element is still in the belief set. Let us consider the three types of justifications.

D2a The justification of a default or an external element E is invalid, iff one or more elements are in the belief set which make it possible to derive reasons pleading against E.

D2b The justification of a derived element E becomes invalid, iff either a reason in its justification becomes invalid, or the justification becomes incomplete.

D3: reason(i) is invalid iff:
a) the justification of rule(i) has become invalid; or
b) rule(i) has been retracted externally; or
c) rule(i) has become irrelevant, because new information or retraction of information makes that its scope conditions are not satisfied anymore; or
d) the justification of one or more of the propositions constituting reason(i) has become invalid;
e) one or more of the propositions constituting reason(i) has been retracted externally.

D4: rl(e) is incomplete, iff since the addition of e to B, one or more elements have been added to or retracted from B, which make it possible to derive reasons for or against e which are not in rl(e).

D5: rl(e) is invalid, iff either it contains an invalid element, or it is incomplete.

D6: If rl is not invalid, it is valid.

A reason for a belief element may have become invalid, while the remaining reasons are sufficient to maintain the belief element. This does not safeguard the justification from becoming invalid. The same counts when a justification has become incomplete, while the new reasons do not change the balance of reasons.

Next to the notion of validity, the weaker notion of loose validity can be defined. The definition makes use of the notions of strictly invalid reasons and justifications.

D7: reason(i) is strictly invalid iff:
a) rule(i) has been retracted; or
b) rule(i) has become irrelevant, because new information or retraction of information makes that its scope conditions are not satisfied anymore; or
c) one or more of the propositions constituting reason(i) have been retracted.

D8: rl(e) is strictly invalid, iff it contains a strictly invalid element.

D9: If rl(e) is not strictly invalid, it is loosely valid.

The distinction between standard and loose validity rests upon the depth of search which is undertaken to establish the validity of a belief element's justification. In the case of (standard) validity, the justifications of elements of justifications are also investigated. This is a recursive procedure which only ends with external or default belief elements. In the case of loose validity, this recursion does not take place.

The recursion which seems to result from the demand that the rules underlying justification elements are relevant is avoided by the assumption that scope conditions are satisfied by default.

Retracting belief elements

A belief element is retracted from the belief set as soon as it is established that its justification is invalid. This implies that belief elements with a strictly invalid justification are also retracted.

3.6. Inference

Satisfaction of compound propositions

The belief set does not contain compound propositions. The satisfied-predicate functions as replacement. This predicate can be applied to ordered pairs, consisting of a proposition and its justification. It is defined as follows:

D10a: satisfied(p, p), iff p/j n B and j is valid.

D10b: satisfied(conj([p_m … p_n]), [j_m…j_n]), iff all propositions p_m … p_n are satisfied, with respectively the justifications j_m … j_n.

D10c: satisfied(or([p_m … p_n]), j_i), iff at least p_i of the propositions p_m … p_n is satisfied, with j_i as justification.

Because of definition 10b, it is possible that a list of justifications appears as an element in another list of justifications. In that case the brackets of the member element list are removed. Moreover, duplicate elements in justification lists are dropped. This procedure corresponds to the fact that the final justification list represents the union of the sets of reasons which constitute the justifications of the conjuncts.

Note that being satisfied implies having a valid justification. In this respect being satisfied is more restricted than being an element of the belief set.

Satisfiability of compound propositions

The satisfiable-predicate is in some cases a replacement for the satisfied-predicate. It is defined as follows:

D11a: satisfiable(p,p), iff p is a derivable element of B.

D11b: satisfiable(conj([p_m … p_n]), [j_m … j_n]), iff all propositions p_m … p_n are satisfiable, with respectively the justifications j_m … j_n.

D11c: satisfiable(or([p_m … p_n]), j_i), iff at least p_i of the propositions p_m … p_n is satisfiable, with j_i as justification.

From these definitions it follows that every proposition which is satisfied, is also satisfiable.

Satisfaction by failure

Just like satisfiability, satisfaction by failure (sbf) is a weaker replacement for satisfaction. It is defined as follows:

D12a: sbf(p,p), iff not ~p n B with a valid justification.

D12b: sbf(conj([p_m … p_n]), [$j_{m.}$..j_n]), iff all propositions p_m … p_n are satisfied by failure, with respectively the justifications j_m … j_n.

D12c: sbf(or([p_m … p_n]), j_i), iff at least p_i of the propositions p_m … p_n is satisfied by failure, with j_i as justification.

Satisfaction, satisfiability, relevance, and (strict) applicability of rules

If the conditions of a rule are satisfied, the rule itself is also satisfied. The justification of the condition also applies to the rule.

D13: satisfied(rule(i_1, p_1, p_2, i_2), j) iff satisfied(p_1, j).

If the conditions of a rule are satisfiable, the rule itself is also satisfiable. The justification of the condition also applies to the rule.

D14: satisfiable(rule(i1, p_1, p_2, 2), j) iff satisfiable (p_1, j).

If the scope conditions of a rule are satisfied by failure, the rule itself is relevant. The scope conditions of a rule are represented by a meta-rule, indicating under what conditions the use of the standard rule is excluded. If such a meta-rule is in B with a valid justification, and if it is strictly applicable (this notion is defined right away), the scope conditions are not satisfied. Otherwise they are.

D15: relevant(rule(i_1)), iff no rule(i_2, p, excludes(i_1)) is in B with a valid justification, such that s_applicable(rule(i_2)).

S-applicable in the preceding definition stands for strictly applicable. In order for a rule to strictly apply, it must both be satisfied and relevant. This leads us to the following definition of strict rule applicability:

D16: s_applicable(rule(i)), iff satisfied(rule(i), j) and relevant(rule(i)).

A somewhat looser variant is just applicability. In order for a rule to apply, it must both be satisfiable and relevant. This leads us to the following definition of rule applicability:

D17: applicable(rule(i)), iff satisfiable(rule(i), j) and relevant(rule(i)).

Deriving standard reasons from rules

The following rule indicates how reasons are derived from rules:

R1a: reason(i_1, p_1, p_2, i_2) can be derived from rule(i_1, p_1, p_2, i_2), iff rule(i_1) is in B with a valid justification, and applicable(rule(i_1)).

R1b: reason(i_1, p_1, excludes(i_2)) can be derived from rule(i_1, p_1, excludes(i_2)), iff rule(i_1) is in B with a valid justification, and s_applicable(rule(i_1)).

Note that in the rule for standard reasons the looser notion of applicability is used, and not strict applicability. If the latter notion were used, the potential number of reasons for or against a conclusion would diminish.

If a reason for or against a conclusion can be derived, this reason is said to be a reason for, respectively against that conclusion.

Weighing reasons
In order to determine whether a conclusion can be derived, all reasons pleading for or against that conclusion must be weighed. Since reasons pleading against a conclusion have a negative weight, the weighing of the reasons comes down to summing them. To determine the weight necessary to draw positive, respectively negative conclusions about a proposition, threshold values must be set. These are called T_{max} and T_{min}, for respectively the bottom-threshold for positive conclusions and the ceiling-threshold for negative conclusions.

The following definitions precisify this:

D18: rl(e) denotes the set of all reasons pleading for or against e.

D19: ws(e) is the sum of the weight arguments of all reasons n rl(e).

D20: T_{max} and T_{min} are integers, where $T_{max} >= T_{min}$.

A conclusion can be derived according to the following rules:

R2a: It is possible to derive e/ rl(e) iff ws(e) > T_{max}.

R2b: It is possible to derive ~e/ rl(e) iff ws(e) < T_{min}.

If a conclusion is derived from the belief set, the proposition or rule, together with its justification is added to the belief set.

3.7. Adding to and retracting from the belief set

There are two ways to add elements to the belief set. One way is addition from outside. This topic falls outside the scope of this paper. The other way is addition as a result of derivation (application of rules R2a or R2b).

Adding a new belief element may have consequences for the justifications of other belief elements and indirectly even for the justification of the added element itself. These consequences are not drawn directly, but only when the validity of an elements justification is checked while making new derivations. The belief set is not updated continuously.

There are two reasons for not continuously updating the belief set. One reason has to do with the limited information processing capabilities of humans. Humans are not able to adapt all of their knowledge to local changes. Where RBR aims at reflecting the nature of human information processing, this human shortcoming is adopted. The other reason has to do with computability and efficiency aspects in case RBR is implemented in a computer program.

There are also two ways to retract elements from the belief set. The first way is retraction from outside, which is not discussed in this paper. The second way, or - better - occasion, is when the validity of a belief element's justification is checked when that element is used in making derivations. For propositions this is the case when definitions 10a and 11a are applied, and for rules when rules R1a and R1b are applied. If the justification turns out to be invalid, the belief element is retracted from B.

Retracting a belief element may have consequences for the justifications of other belief elements. Just as with additions, these consequences are not drawn directly, but only when the validity of an elements justification is checked while making new derivations.

4. Formalized conflict rules

To show how RBR can be put to practical use, I will formalize the example of the conflicting traffic regulations. Let $T_{max} = T_{min} = 0$, and let B contain the following elements:

P_1/2: The traffic light is red.
P_2/2: The police officer signals to drive on.
P_3/2: rule(1) is a lex specialis with regard to rule(2). (lex_specialis(rule(1), rule(2)))

rule(1, P_2, P_4, -2)/ default
rule(2, P_1, P_4, 3)/ default
rule(3,
lex_specialis(rule(i_1,p_1,p_3,i_3), rule(i_2,p_2,p_3,-i_4)),
rule(i_5,p_1, excludes(rule(i_2))),
100)/ default.
(If one rule is a lex specialis with regard to another rule, this is a reason with the weight 100 to accept a meta-rule which makes the applicability conditions of the first rule an exclusionary reason with regard to the second rule.)

(The formalization of this rule, which violates syntactical requirements formulated above, indicates the need to extend RBR from a propositional logic-like theory to a predicate logic-like theory.)

P_4 stands for 'The driver must stop.'

Assume we try to derive P_4. Because of rule(1) and P2, we have reason(1, P_2, P_4, -2). If we also had reason(2, P_1, P_4, 3), ws(P_4) would have been 1. Since T_{max} is 0, it would have been possible to add P_4 / [reason(2, P_1, P_4, 3), reason(1, P_2, P_4, -2)] to B.

However, since P_3 and rule(3)   B, and since the justifications of these belief elements are not invalid, there is a reason(3, P_3, rule(4), 100). There are no conflicting reasons, and therefore rule(4, P_2, excludes(rule(2))/ [reason(3, P_3, rule(4), 100)] is derivable from B.

Now there are two possibilities. First let us assume that rule(4) is actually derived from, and added to B. Since P_2 B with a valid justification and rule(4) is relevant, rule(4) is strictly satisfied. Consequently rule(2) is not relevant and therefore not applicable. As a consequence it is not possible to derive reason(2, P_1, P_4, 3), with the effect that the weight of reason(1) is not compensated. The result of the weighing of reasons is that ws(P_4) = -2. Since -2 < T_{min}, ~P_4/[reason(1)] is added to B.

It is also possible that rule(4) is not actually derived from B. Although it is derivable, it is not in B. Therefore rule(2) is relevant, although rule(4) is strictly applicable. Now we have reason(2, P_1, P_4, 3) and we can derive P_4.

It turns out that the order in which belief elements are derived is important for the possibility to derive them. This should not be a problem since RBR models a process and not semantic content.

5. Concluding remarks

In the course of this paper, sometimes related notions of different strength are defined. Examples are the notions of satisfaction, satisfaction by failure, and satisfiability and the notions of validity, loose validity and strict validity. I made use of the distinctions between the first three notions, but did not use the distinction between the latter three in this paper.

By combining the different variants of the notions in various ways, it is possible to obtain several variants of RBR. The reason why I presented different variants of validity without actually using them, is to give a first indication of the directions into which RBR can be elaborated. Possible elaborations especially involve different depths of search to maintain belief set 'consistency'.

The deeper the search, the more 'consistent' the belief set, and the more 'rational' the procedures of belief revision. But also: the deeper the search, the bigger the chance to be entangled in circularities. And one can also ask whether it is always rational to search as deep as possible.

Clearly the research on RBR is not finished yet. It is not only necessary to develop a dialogical version, RBR must also be elaborated to take (conflicting) modalities into account. Moreover, the topic of possible circularities in derivations must be investigated, just as the topics of meta-rules influencing the weight of reasons, and the intervention in the belief set from outside.

Despite this list of topics for research, the basis of RBR as presented in this paper fits in better with actual legal reasoning than classical logics do. Especially the distinctions between propositions and rules, between standard and meta-rules and -reasons, and between ordinary and scope conditions of rules seem fruitful. That makes the endeavour to build on the RBR basis a promising one.

6. Acknowledgements

The author wishes to thank Henry Prakken and Kees de Vey Mestdagh for their valuable comments on an earlier version of this paper. He hopes that the resulting changes turn out in majority to be improvements.

7. References

Doyle, J. (1979). A Truth Maintenance System, Artificial Intelligence, vol. 12, p. 231-272.

Geach, P. and Black, M. eds. (1962). Translations from the Philosophical Writings of Gottlob Frege, Basil Blackwell, Oxford.

Haack, S. (1978). Philosophy of Logics, Cambridge University Press, Cambridge e.a.

Hage, J.C. (1987a). Feiten en betekenis; een verhandeling over ontologie en praktische rede, Leiden.

Hage, J.C. (1987b). De betekenis van niet-standaardlogica's voor juridische expertsystemen. Computerrecht 1987/4, p. 233-239.

Hage, J.C. (1991). An Alternative for Deontic Logic. Legal Knowledge Based Systems. Aims for Research and Development, (C. van Noortwijk, A.H.J. Schmidt en R.G.F. Winkel eds.), Koninklijke Vermande BV, Lelystad, p. 59-69.

Kleer, J. de (1986). An assumption-based TMS, Artificial Intelligence, vol. 12, p. 127-162.

Lorenzen, P. and Lorenz, K. (1978). Dialogische Logik, Wissenschafliche Buchgesellschaft, Darmstadt.

Ross, W.D. (1930). The Right and the Good, Clarendon Press, Oxford.

Searle, J. (1978). Prima Facie Obligations. J. Raz (ed.), Practical Reasoning, Oxford University Press, Oxford.