Chapter 4

Formalizing rules: a comparative survey

In the chapters 2 and 3, we have described our approach to formalizing rules: Reason-Based Logic. In this chapter, we discuss a number of other approaches, and compare them to ours. We focus on issues concerning rules that arise because of the defeasibility of arguments.[1]

In section 1, we make some general remarks on rules and their role in argumentation. In section 2, we treat the classic formalization of rules as material conditionals, and to what extent this formalization can cope with a number of issues related to the defeasibility of arguments. Section 3 continues with a discussion of approaches to dealing with the relevance of rule conditions for rule conclusions. We discuss approaches to dealing with exceptions to rules in section 4, and approaches to dealing with rule conflicts in section 5. In section 6, we look at reasoning about rules.

We wish to stress that many of the observations in this chapter are not original.[2] However, we have added some originality by focusing on different issues instead of on specific formalisms. We have selected a number of well-known and influential formalisms, and use them to explain general approaches to the issues. In this way, the approach to formalizing rules of Reason-Based Logic is put in perspective.

1 Rules in argumentation

In this section, we explain our view on rules. We start with the relation between rules and arguments. Some remarks on syllogistic and enthymematic arguments follow. The section ends with a discussion of ordinary rule application.

---

1 Nute (1980) and Sanford (1989) describe other interesting topics, such as counterfactual conditionals.

2 We have especially benefited from the discussions by Haack (1978), Prakken (1993a, chapters 5 and 7) and Makinson (1994).

1.1 Rules and arguments

We recall our interpretation of rules and their relation to arguments (see also chapter 1, section 4.1 and chapter 2, section 1.1). As our starting point, we take informal arguments as they occur in practice, e.g.,

Mary is born in Maastricht.
So, Mary pronounces the letter g softly.
So, people can tell that Mary is from the south of the Netherlands.

We present arguments in an idealized form, with clearly distinguished steps. Each step consists of a reason and a conclusion, as follows:

Reason.
So, *Conclusion*.

Arguments can consist of several steps. In that case, the conclusion of one step is the reason of the next. The example argument consists of two steps. The first step has the reason ‘Mary is born in Maastricht’ and the conclusion ‘Mary pronounces the letter g softly’, the second step the reason ‘Mary pronounces the letter g softly’ and the conclusion ‘People can tell that Mary is from the south of the Netherlands’.

The steps in the argument can also occur in other arguments. For instance, the first step in the argument above also occurs in the following argument:

Mary is born in Maastricht.
So, Mary pronounces the letter g softly.
So, people from Amsterdam may find Mary’s accent amusing.

In other words, steps in an argument are independent of the particular argument in which they occur. Each step can be used in an argument because there exists some relation between the reason and the conclusion of the step. This relation between reason and conclusion as expressed by the argument step, is what we call a *rule*.

Often argument steps follow a pattern. For instance, the first argument above can be made for anyone who is born in Maastricht. We have the following argument scheme:

Person is born in Maastricht.
So, *Person* pronounces the letter g softly.
So, people can tell that *Person* is from the south of the Netherlands.

The steps in the argument scheme can be used in an actual argument independently of the particular person mentioned. *Person* is a variable, that can be filled in at will: whoever the person *Person* is, Mary, Peter, or Fred, the scheme gives rise to an acceptable argument. Also the relation between reason and conclusion in a step in such an argument scheme is called a rule, but this time it is a rule with a variable.

There are few things about rules of reasoning that are generally agreed upon. However, a common starting point is that a rule has a condition and a conclusion. The condition and the conclusion of a rule correspond to the reason and the conclusion in an argument step, respectively. So, an argument step of the form

Reason.
So, *Conclusion*.

corresponds to a rule with condition *Reason* and conclusion *Conclusion*. It may seem inconsistent terminology to use two terms, ‘reason’ and ‘condition’ for corresponding things. However, there is a difference: if the condition of a rule is used as a reason in an argument, the reason is assumed to hold, while for the validity of a rule it is irrelevant whether its condition holds.

1.2 Syllogistic and enthymematic arguments

If in introductory texts on classical deductive logic examples of informal arguments are given, they typically look as follows (e.g., Purtill, 1979; Copi, 1982, especially p. 235 ff.):

1. John is a thief. If John is a thief, then he should be punished.
   So, John should be punished.

2. Either John is married to Mary or John is married to Edith. John is married to Mary.
   So, John is not married to Edith.

They are used to introduce logical connectives, such as ‘If ... then ...’ and ‘Either ... or ...’. In ordinary language, one also finds the following, closely related arguments that do not contain these connectives:

1’. John is a thief.
So, John should be punished.

2’. John is married to Mary.
So, John is not married to Edith.

These arguments result from the arguments 1 and 2 above by omitting one of the premises. From the point of view of classical logic, the first two arguments are complete, while in the second two one of the premises is missing. The arguments 1’ and 2' are called *enthymematic*, in contrast with their *syllogistic* counterparts 1 and 2, that explicitly contain all premises (Copi, 1982, pp. 235, 253).[3]

In this thesis, we have given examples of arguments that resemble the syllogistic type of argument and of arguments that resemble the enthymematic type. This may seem inconsistent. However, the apparent inconsistency disappears if it is noted that the distinction between syllogistic and enthymematic arguments only has meaning *relative to a set of rules*. For instance, the syllogistic arguments above are complete, relative to the rules (or rule schemes) Modus Ponens and Disjunctive Syllogism underlying the argument schemes:

> $State\_of\_affairs_1$. If $State\_of\_affairs_1$, then $State\_of\_affairs_2$.
> So, $State\_of\_affairs_2$.

> Either $State\_of\_affairs_1$ or $State\_of\_affairs_2$. $State\_of\_affairs_1$.
> So, not $State\_of\_affairs_2$.

Relative to these rules, we can distinguish the syllogistic arguments 1 and 2, in which all premises are explicitly stated, and the enthymematic arguments 1' and 2', in which one or more premises are missing.

The example arguments 1' and 2', that are enthymematic with respect to Modus Ponens and Disjunctive Syllogism, are syllogistic with respect to the rules that underlie the argument schemes

> *Person* is a thief.
> So, *Person* should be punished.

and

> $Person_1$ is married to $Person_2$.
> So, $Person_1$ is not married to $Person_3$.[4]

Clearly, our interpretation of rules is closely related to the warrants in Toulmin's (1958) argument scheme.[5]

We have taken some effort to state our interpretation of the notion 'rule' as clearly as possible, for two reasons. First, we think that research on the formalization of reasoning with defeasible arguments should be thoroughly

---

[3] The distinction between syllogistic and enthymematic arguments was already made by Aristotle (cf. Copi, 1982).

[4] It is sometimes objected that the rules underlying these arguments refer to the meaning of the phrases used. This ignores the fact that also a rule such as Modus Ponens refers to the meaning of its phrases, namely the meaning of 'If ..., then ...', which as we will see is not uncontroversial.

[5] Toulmin's argument scheme has recently inspired several researchers (cf., e.g., Bench-Capon, 1995).

grounded in intuitions, simply because that research is inspired by the intuitive differences between actual reasoning and, for instance, deductive reasoning. This is in line with our general method of research (chapter 1, section 7)

Second, different intuitions can cause much confusion. Therefore, we stress that our interpretation of rules differs from several other interpretations in the literature, such as rules of inference, material conditionals, or default rules. Indeed, there is no single, generally accepted interpretation of the notion ‘rule’. In fact, a significant part of the research on defeasible reasoning can be regarded as a search for the meaning, or, better, for different meanings of the notion ‘rule’.

1.3 Ordinary rule application

In any interpretation of rules, they can in some sense be applied: if there is a rule, the condition of which holds, the conclusion of the rule follows. Here ‘holds’ and ‘follows’ can be interpreted in many ways, for instance as ‘be true’, ‘be derivable’, or ‘be justified by an argument’. The latter interpretation will be our intuitive guideline in this chapter.

Since we will be dealing with several different formalisms, a notational convention is useful. If the conclusion *Conclusion* follows from the assumptions $Assumption_1$, $Assumption_2$, ..., $Assumption_n$, we write:

$$Assumption_1, Assumption_2, ..., Assumption_n \mid\!\sim Conclusion$$

Our guiding interpretation of this notation is as follows: assuming $Assumption_1$, $Assumption_2$, ..., $Assumption_n$, the conclusion *Conclusion* is justified (by some argument).

Using this notation, ordinary rule application is denoted as follows:

$$Rule, Condition \mid\!\sim Conclusion$$

Here *Rule* denotes that there is a valid rule that has *Condition* as its condition and *Conclusion* as its conclusion.

In First-Order Predicate Logic (see, e.g., Van Dalen (1983) or Davis (1993)), there is an obvious candidate to formalize rules, namely the material conditional.[6] A rule with condition *Condition* and conclusion *Conclusion* can be represented as the material conditional $Condition \rightarrow Conclusion$, and ordinary rule application can be interpreted in two well-known (and equivalent) ways, namely semantically and proof-theoretically:

If $Condition \rightarrow Conclusion$ and *Condition* are true, then *Conclusion* is true.
From $Condition \rightarrow Conclusion$ and *Condition*, *Conclusion* is derivable.

---

[6] The material *conditional* is often called the material *implication*. Sanford (1989), joining Quine, explains why this is uncareful use of language.

These are usually formally represented as follows:

Condition $\rightarrow$ *Conclusion*, *Condition* $\models$ *Conclusion*
Condition $\rightarrow$ *Conclusion*, *Condition* $\vdash$ *Conclusion*

In our notational convention, both become:

Condition $\rightarrow$ *Conclusion*, *Condition* $\mid\!\sim$ *Conclusion*

We stress that the symbol $\mid\!\sim$ does not give preference to a semantically or a syntactically defined consequence relation.

In the chapters 2 and 3, we discussed another candidate to formalize rules, namely the rule of Reason-Based Logic. In comparison with the complexity of the rule of Reason-Based Logic, the material conditional is attractively simple. Therefore an important question arises. Why is the material conditional approach to rules unsatisfactory? That is the subject of the next section.

2 Rules as material conditionals

In this section, we discuss the material conditional approach to rules. First we discuss the relevance of rule conditions for rule conclusions and the paradoxes of the material conditional. Then we discuss the behavior of the material conditional with respect to exceptions and conflicts. The section ends with a discussion of the problems of the material conditional related to reasoning about rules.

2.1 Relevance and the paradoxes of the material conditional

If we formalize rules as material conditionals, the first problems that we encounter concern the relevance of the condition for the conclusion.

The rule of our example above, that allowed the argument steps of the scheme

Person is born in Maastricht.
So, *Person* pronounces the letter g softly.

shows the relevance of the condition of a rule for its conclusion. The fact that someone is born in Maastricht is *relevant* for the fact that someone pronounces the letter g softly, in the sense that under normal circumstances the second follows *because* the first holds. This relevance is a consequence of the way the world is: people born in Maastricht, normally pronounce the letter g softly. As a result, the demand of the relevance of a rule's condition for its conclusion is in principle a matter of the domain theory.

For instance, a domain theory that contains a rule with condition 'The sky is blue' and conclusion 'Amsterdam is the capital of the Netherlands' does not meet the relevance demand. However, the relevance demand is not only a matter of the domain theory, but also of the allowed inferences. We show this using the material conditional as an example. It turns out that material conditionals have properties that are not in line with the relevance demand.

For instance, if we assume that Mary is *not* born in Maastricht, the material conditional with condition Mary_is_born_in_Maastricht and conclusion Mary_pronounces_the_letter_g_softly follows:

¬Mary_is_born_in_Maastricht |~ Mary_is_born_in_Maastricht →
Mary_pronounces_the_letter_g_softly

In fact, any material conditional with condition Mary_is_born_in_Maastricht follows, for instance:

¬Mary_is_born_in_Maastricht |~ Mary_is_born_in_Maastricht →
¬Mary_pronounces_the_letter_g_softly
¬Mary_is_born_in_Maastricht |~ Mary_is_born_in_Maastricht →
There_is_life_on_Mars
¬Mary_is_born_in_Maastricht |~ Mary_is_born_in_Maastricht →
¬Mary_is_born_in_Maastricht

The examples have been chosen in such a way that the conditions of the material conditionals become decreasingly relevant for their conclusions. Interpreted as rules that give rise to acceptable arguments, these material conditionals become increasingly absurd. For instance, in our interpretation, the last example reads as follows. Assuming that Mary is not born in Maastricht, there is a rule that makes the argument

Mary is born in Maastricht.
So, Mary is not born in Maastricht.

acceptable.

These examples are due to the first of the following so-called paradoxes of the material conditional (cf., e.g., Haack, 1978, p. 37):

$\neg A \mid\!\sim A \rightarrow B$
$B \mid\!\sim A \rightarrow B$
$\mid\!\sim (A \rightarrow B) \vee (B \rightarrow A)$

Examples of the second are:

Mary_pronounces_the_letter_g_softly |~ Mary_is_born_in_Amsterdam →
Mary_pronounces_the_letter_g_softly

Mary_pronounces_the_letter_g_softly |~ There_is_life_on_Mars →
Mary_pronounces_the_letter_g_softly

Interpreting the latter, we find: assuming that Mary pronounces the letter g softly, there is a rule that makes the argument

There is life on Mars.
So, Mary pronounces the letter g softly.

acceptable.

An example of the third paradox is:

|~ (There_is_life_on_Mars → Mary_pronounces_the_letter_g_softly) ∨
(Mary_pronounces_the_letter_g_softly → There_is_life_on_Mars)

Interpreting this, we find that there is either a rule that makes the argument

There is life on Mars.
So, Mary pronounces the letter g softly.

acceptable, or a rule that makes the argument

Mary pronounces the letter g softly.
So, there is life on Mars.

acceptable.

The examples show that the material conditional does not behave well with regard to relevance. Even if we are careful and assume only material conditionals which have conditions that are relevant for their conclusions, we obtain many other material conditionals for free which lack that property. This has been recognized for long, and is generally considered a drawback of the formalization of rules as material conditionals. For instance, the paradoxes of the material conditional led C.I. Lewis to the definition of the strict conditional (that turned out to have similar paradoxes of its own),[7] and Anderson and Belnap to the development of their logic of relevance.[8]

Some approaches to dealing with relevance are discussed in section 3.

---

7 Cf. Haack (1978, p. 37) and Sanford (1989, p. 68ff.).
8 Cf. Haack (1978, p. 37, p. 198ff.) and Sanford (1989, p. 129ff.).

2.2 Exceptions to rules

Another source of problems for the material conditional are exceptions to rules. We have already seen several examples of exceptions in the previous chapters (chapter 1, section 4.1, chapter 2, section 1.2, chapter 3, section 5).

There are two intuitive requirements for reasoning with rules with exceptions:

STANDARD CASE

If there is a rule the condition of which holds, then the rule's conclusion follows.

EXCEPTION CASE

If there is a rule the condition of which holds, and there is an exception to the rule, then the rule's conclusion does not follow.[9]

If we model rules as material conditionals, we get the following:

STANDARD CASE

Condition, *Condition* $\rightarrow$ *Conclusion* $\mid\!\sim$ *Conclusion*

EXCEPTION CASE

Condition, *Condition* $\rightarrow$ *Conclusion*, *Exception* $\not\mid\!\sim$ *Conclusion*

The latter is clearly false. We recall the property called monotonicity:

If *Assumptions* $\mid\!\sim$ *Conclusion*,
then *Assumptions*, *More_assumptions* $\mid\!\sim$ *Conclusion*.

It follows immediately that a reasoning formalism that meets the two requirements above cannot be monotonic. Since First-Order Predicate Logic is monotonic, we conclude that reasoning with rules with exceptions cannot be represented in it.

It may at first seem strange, but the requirement in the standard case, makes reasoning with rules with exceptions nonmonotonic, and not the requirement in the exception case. In the standard case, one jumps to the conclusion of the rule, while there might be an exception. It would be more careful to add the assumption that there is no exception, as follows:

CAREFUL STANDARD CASE

If there is a rule the condition of which holds, and there is no exception, then the rule's conclusion follows.

Clearly, this careful requirement does not lead to nonmonotonicity. For instance, the deductive consequence relation of Reason-Based Logic (chapter 2, beginning of

9 Of course, the rule's conclusion can hold, as a result of *other* information.

section 6) is careful in this sense. However, as we already discussed there, this carefulness leads to a weak notion of consequence.

Other approaches to dealing with exceptions to rules are discussed in section 4.

2.3 Rule conflicts

A third source of problems for the formalization of rules as material conditionals are rule conflicts. We have already seen several examples in the previous chapters (chapter 1, section 4.2, chapter 2, section 1.3, chapter 3, section 6). We mention two types of unwanted behavior of the material conditional.

The first type of unwanted behavior is that, if there is a conflict of material conditionals, i.e., their conclusions are incompatible and their conditions satisfied, anything follows. Formally,

$$Condition_1, Condition_2, Condition_1 \rightarrow Conclusion, Condition_2 \rightarrow \neg Conclusion \mid\!\sim Anything$$

For instance, interpreting rules as material conditionals, we find: if thieves are punishable, minor first offenders are not punishable, and John is a minor thief, then Fermat's theorem is true. This easy way of settling Fermat's theorem is of course useless since we can also conclude that it is false. Clearly, this behavior of the material conditional is unwanted if one accepts the existence of rule conflicts. Intuitively, a conflict of rules should not lead to a contradiction from which anything follows. We have the following intuitive property:

RULE CONFLICT

If there are rules with incompatible conclusions, the conditions of which hold, no contradiction follows.

The second type of unwanted behavior of the material conditional occurs even if the conditions of rules with incompatible conclusions are not satisfied. We have the following:

$$Condition_1 \rightarrow Conclusion, Condition_2 \rightarrow \neg Conclusion \mid\!\sim Condition_2 \rightarrow \neg Condition_1$$

For instance, if thieves are punishable and minor first offenders are not, then minor first offenders are not thieves. It would be very nice for governments if simply announcing that minor first offenders are not punishable would have this effect. Intuitively, it is unwanted that rules with incompatible conclusions lead to other rules, as naively as above. The property is related to the property of the so-called contraposition of the material conditional:

$$Condition \rightarrow Conclusion \mid\!\sim \neg Conclusion \rightarrow \neg Condition$$

This property can easily lead to strange results. For instance, if we have that suspects are presumed innocent, do we also have that those who are not presumed innocent are not suspect?

Both types of unwanted behavior show that rules easily allow for too many conclusions. First, we saw that a conflict of rules should not lead to a contradiction; second, that a rule should not lead to its contraposition.

This is opposite to the situation in the case of exceptions, where we saw that rules sometimes allow too few conclusions: in the standard case, we want to jump to a conclusion, even if there might be an exception.

In Figure 1, the tension between too few and too many conclusions is suggested. The set of strict conclusions that follow from a set of assumptions is often considered too small. As a result, one wants to enlarge that set by allowing tentative conclusions. On the other hand, if one enlarges the set too much, the boundary of consistency is crossed.[10] Since this is also unwanted, one wants to constrain the set of tentative conclusions, in order to maintain consistency.

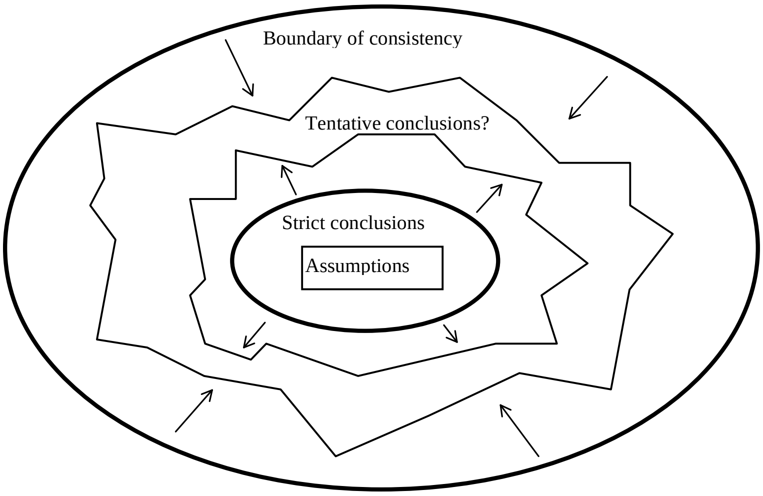

Figure 1: The tension between too few and too many conclusions

As the figure shows, an acceptable set of tentative conclusions that follow from a set of assumptions includes the set of strict conclusions, and is included in some consistent set.

Other approaches to dealing with rule conflicts are discussed in section 5.

[10] The figure suggests that there is a clear, unique, boundary of consistency. This is of course not the case: there can be many different maxiconsistent sets. However, this is unessential for what the figure attempts to depict.

2.4 Reasoning about rules

As a fourth source of problems for formalizing rules as material conditionals, we discuss reasoning about rules. We distinguish two types of reasoning about rules: reasoning with rules as conclusions, and reasoning that involves facts about rules.[11]

Assume that we consider the arguments

It is raining and I did not bring a rain coat.
So, my clothes get wet.

and

My clothes get wet.
So, I will feel uncomfortable.

to be acceptable. It seems reasonable to conclude that also the argument

It is raining and I did not bring a rain coat.
So, I will feel uncomfortable.

is acceptable. As a result, the following argument, in terms of the rules that give rise to these arguments, is acceptable:

‘If it is raining and I did not bring a rain coat, my clothes get wet’ is a valid rule.
‘If my clothes get wet, I feel uncomfortable’ is a valid rule.
So, ‘If it is raining and I did not bring a rain coat, I will feel uncomfortable’ is a valid rule.

This argument is an example of reasoning about rules, in which the conclusion of the argument is a rule. Other examples have facts about rules as their conclusion. There can be an argument concerning exceptions, e.g.,

John is driving on a German highway.
So, there is an exception to the rule ‘If John drives faster than 120 kilometers per hour, he can be fined’.

or priority relations between rules, e.g.,

John knows Mary well.
Alex hardly knows Mary.
So, the rule ‘If John says Mary is nice, then Mary is nice’ prevails over the rule ‘If Alex says Mary is not nice, then Mary is not nice’ in case of a conflict.

[11] We will later see (section 6) that in Reason-Based Logic this distinction disappears.

If rules are formalized as material conditionals, the first type of reasoning about rules, in which a rule occurs as a conclusion, can apparently be dealt with. For instance, the first example we gave corresponds to the following property of material conditionals, called transitivity:

$$A \to B, B \to C \mathrel{|\!\!\sim} A \to C$$

However, this can hardly be considered as reasoning about rules, since it is not based on information about the particular rules involved. Transitivity is a property that holds for general material conditionals, and does not depend on any particular information for particular material conditionals.

Moreover, rules do not always have the property of transitivity. A counterexample is the following. Assume we have the two argument schemes:

Person lives in Curaçao.
So, *Person* is Dutch.

and

Person is Dutch.
So, *Person* lives in Europe.

Even if these arguments are acceptable, the argument scheme

Person lives in Curaçao.
So, *Person* lives in Europe.

need not be acceptable, since Curaçao is in the Caribbean region, and not in Europe. The fact that the property of transitivity does not hold for the arguments in this case is the result of the fact that the rule ‘If someone is Dutch, he lives in Europe’ can have exceptions. Since material conditionals have the property of transitivity, the rules underlying the example arguments cannot be formalized as material conditionals.

For the other type of reasoning about rules, involving facts about rules (e.g., about exceptions, conflicts or priorities), modeling rules as a material conditional is clearly inadequate, since this would require that it is possible to express facts about material conditionals in the object language. This is not possible in standard First-Order Predicate Logic.

Other approaches to dealing with reasoning about rules are discussed in section 6.

3 Relevance

In order to avoid the problems of the material conditional with regard to relevance, a special syntactic form should be used, reserved for the representation of rules. In this way, it is possible to specify the properties of rules from scratch.

We discuss three approaches that follow this idea. The first is to fixate the set of rules. The second is to treat rules as special sentences. The third is to treat rules as special objects.

3.1 Fixating a set of rules

As an example of the first type of approach, in which the set of rules is fixated, we discuss Reiter's Default Logic (Reiter, 1980, 1987). We start with a summary of his definitions.

Reiter's Default Logic uses the language of First-Order Predicate Logic; for simplicity we use that of Propositional Logic here. The assumptions are encoded as a pair of sets (F, Δ), where F is a set of sentences and Δ is a set of default rules. Such a pair of sets (F, Δ) is called a *theory*.

A *default rule* has the form

$\alpha : \beta_1, \beta_2, ..., \beta_n / \gamma$,

where α, β_1, β_2, ..., β_n, and γ are sentences. Here α is the prerequisite of the default rule, β_1, β_2, ..., β_n are the justifications of the rule, and γ is the consequent of the default rule. Representing a rule as a default rule, the condition of a rule corresponds to the prerequisite of a default rule, and the conclusion of a rule to the consequent of the default rule. The role of the justifications of a default rule is discussed in section 4.2.

An *extension* of a theory (F, Δ) is a set of sentences E, such that $E = E_0 \cup E_1 \cup E_2 \cup E_3 \cup ...$, where

$E_0 = F$, and
$E_{i+1} = Th(E_i) \cup \{ \gamma \mid$ there is an $\alpha : \beta_1, \beta_2, ..., \beta_n / \gamma \in \Delta$, such that $\alpha \in E_i$, and for all j: $\neg\beta_j \notin E \}$ for any $i \geq 0$.[12]

The definition of the E_i depends on E. Intuitively, the definition of an extension makes use of E as an advance guess of the consequences of a theory (F, Δ), and then checks whether this guess can be gradually constructed using the default rules in Δ starting from the fixed information F.[13]

---

[12] For a set of sentences S, Th(S) denotes the set of logical consequences of S in Propositional Logic.

[13] The same technique was used in the definition of the nonmonotonic consequence relation of Reason-Based Logic (chapter 2, section 6).

There is an equivalent fix-point definition of extensions: E is an extension if E = Γ(E), where the operator Γ is defined as follows. Let S be a set of sentences. Then Γ(S) is the smallest set Γ of sentences, such that:

F $\subseteq \Gamma$, and
Γ = Th(Γ), and
For all $\alpha : \beta_1, \beta_2, ..., \beta_n / \gamma \in \Delta$: If $\alpha \in \Gamma$ and for all j: $\neg\beta_j \notin$ S, then $\gamma \in \Gamma$.

(For all S, there is a smallest set with these three properties: it is the intersection of all sets for which the properties hold.)

Not all default theories (F, Δ) have an extension, and if a default theory has an extension, it is not necessarily unique. A sentence that is an element of all extensions of a default theory is said to follow *skeptically* from the theory; a sentence that is an element of (at least) one of the extensions follows *credulously*.

Reiter's starting point is the incompleteness of the information that we have about the world. He proposes to use default rules as 'rules for extending an underlying incomplete first-order theory'. Apparently, he thinks of (default) rules as special rules of inference, separate from the other available information. This is reflected in the formalism proposed. A default theory is defined as the combination of two sets: a set of first-order sentences, representing ordinary, but incomplete information about the world, and separately a set of default rules, representing information to extend the incomplete information about the world. Reiter then defines extensions of a default theory as sets of first-order sentences.

We return to our discussion of relevance. Formalizing rules as Reiter's default rules, it is clear that the problems of the material conditional with regard to relevance are solved. Since extensions cannot contain default rules, no default rule can be the consequence of a default theory. As a result, if a default rule has condition that are not considered relevant for their conclusions, it is only a flaw of the default theory.

This is of course a crude way of solving the problems of relevance. The 'advantage' is at the same time one of the main drawbacks of Reiter's Default Logic: there are no provisions whatsoever to represent relations between rules, or to reason about rules (see also section 6).

3.2 Rules as special sentences

The second approach is less crude than the first, and treats rules as special sentences. The logical language is extended with a special connective to represent rules, as in conditional logics, as for instance defined by Anderson and Belnap,[14] Nute (1980, 1994) and Delgrande (1988). After extending the language with a rule-representing connective, e.g., >, the properties of the connective are

[14] See, e.g., Haack (1978, p. 198ff.) and Sanford (1989, p. 129ff.).

specified on the meta-level by axioms and rules of inference. Some of them might be:

$\mid\!\sim A > A$
$A > C, B > C \mid\!\sim (A \vee B) > C$
$A > B, (A \wedge B) > C \mid\!\sim A > C$

The choice of such axioms and rules of inference is a delicate matter (which led to a large amount of research), and highly depends on which interpretation of rules one has in mind. For instance,

$A > B \mid\!\sim A \wedge C > B$

should hold for strict rules, but not for rules that can have exceptions.

This approach has the advantage that it is possible to represent not only rules, but also certain relations between them, namely those that can be expressed using other connectives of the logical language, as in $((A > B) \wedge (B > C)) \rightarrow (B > C)$. Of course, the axioms and rules of inference that guide this reasoning must be chosen carefully, in order to meet the demand of relevance. For instance, a rule of inference such as

$A > B \mid\!\sim A \rightarrow B$

could lead to the same unwanted results as with the material conditional, and would therefore probably be a bad choice. However, by carefully choosing axioms and rules of inference, it is in this approach in principle possible to deal with the problems of relevance.

3.3 Rules as special objects

The third approach is to treat rules as special objects, and is used in Reason-Based Logic (chapter 2). Just as in the previous approach, rules can be represented in the logical language. In Reason-Based Logic, they have the form:

rule(*condition*, *conclusion*)

However, there is an important difference with the previous approach: rules are not treated as sentences in the language, but as terms, since rules are considered as special objects. The properties of these rules-as-objects can be represented as sentences of the logical language. For instance, the validity of a rule is expressed as

Valid(rule(*condition*, *conclusion*))

RBL rules also have properties that are specified on the meta-level (described in chapter 2, section 4). For instance, an excluded rule the condition of which is satisfied is not applicable:

Condition, Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) |~
¬Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)[15]

Nevertheless, in comparison with the conditional logic approach, the properties specified on the meta-level leave much room for the specification of the rule properties in the logical language. We come back to this in section 6, where we discuss reasoning about rules.

In Reason-Based Logic this approach has been chosen, because we regard many of the properties of rules as part of the domain theory. This has the advantage that it is possible to represent different types of rules with different properties. For instance, the properties of strict rules are clearly different from those of rules that can have exceptions. In Reason-Based Logic, such properties can flexibly be represented in the domain theory. For instance, a domain theory can be such that the relevance of the rule's condition for its conclusion is implied by the rule's validity. In general, high demands are made on the domain theory.

An alternative approach to represent types of rules with different properties would be to use different syntactic structures for each type of rules. Since the properties are then represented at the meta-level (as discussed in section 3.2), this approach is a little less flexible then the approach discussed here.

4 Exceptions to rules

In this section, we discuss approaches to dealing with rules with exceptions. We do this in two parts. First, we discuss different approaches to the representation of exceptions. Second, we discuss approaches to dealing with exceptions and defeasible reasoning.

4.1 Representing exceptions

We discuss three approaches to the representation of exceptions to rules. The first uses negative rule conditions. The second uses identifiers of rules and a special predicate. The third treats rules as special objects.

[15] Recall that there is a translation from sentences (e.g., *Condition*) to terms (e.g., *condition*), as described in chapter 2, section 4.3.

Negative rule conditions

The first approach to the representation of an exception is as an additional negative condition of a material conditional, as follows:

$$\begin{aligned}&\mathit{Condition}\\&\qquad \wedge \neg \mathit{Exception}\\&\quad \rightarrow \mathit{Conclusion}\end{aligned}$$

There are two drawbacks with representing exceptions as negative conditions. The first is that an additional exception would require a change of the rule itself:

$$\begin{aligned}&\mathit{Condition}\\&\qquad \wedge \neg \mathit{Exception}\\&\qquad \wedge \neg \mathit{Exception}'\\&\quad \rightarrow \mathit{Conclusion}\end{aligned}$$

The second drawback is that there is no formal difference between the condition of a rule and its exceptions. For instance, the material conditional

$$A \wedge \neg B \wedge \neg C \rightarrow D$$

can represent a rule with condition A, conclusion D, and exceptions B and C, but also a rule with condition $A \wedge \neg B$, conclusion D, and exception C.

Both drawbacks conflict with the intuition that a rule is characterized by its condition and conclusion. What we would like is a system in which the existence of an additional exception to a rule is simply an additional fact about that rule.

Rule identifiers and exception predicates

The second approach to the representation of exceptions solves this disadvantage. It is characterized by the use of rule identifiers and a special purpose predicate.[16] A rule is represented as a material conditional, but has an extra condition to represent that it has no exception, for instance as follows:

(∗) $\mathit{Condition} \wedge \neg \mathrm{Exception}(\mathit{identifier}) \rightarrow \mathit{Conclusion}$

Different rules should have different identifiers. Exceptions can now be represented as follows:

---

[16] The use of exception predicates stems from the early days of the research on nonmonotonic logics. Prakken (1993a, p. 84ff.) gives an extensive overview of different variants of this technique, in different logical formalisms.

(+) *Exception* → Exception(*identifier*)

In this representation, an additional exception does not require a change of (∗), but can be represented as an additional assumption:

Exception' → Exception(*identifier*)

If such a material conditional representing an exception is itself a rule that can have exceptions, this can easily be represented by giving it its own identifier and exception clause. For instance, the material conditional (+) becomes:

Exception ∧ ¬Exception(*identifier2*) → Exception(*identifier*)

The problem with this approach to the representation of exceptions is that it is rather ad hoc. The meaning of 'rule' and 'exception' are unclear and underspecified. For instance, is a material conditional of the form (∗) a rule? But then, what does the identifier of the rule refer to? Maybe the identifier is the rule? Does ¬Exception(*identifier*) imply that there is a rule with the identifier *identifier*? Taking these questions seriously, we arrive at the third approach to the representation of exceptions.

Rules as special objects

The third approach to the representation of exceptions is to treat rules as special objects that can have properties. One of the properties of a rule can be that there is an exception to the rule. So, the existence of an exception to a rule is considered as a fact about the rule. Additional exceptions do not change the rule itself, but are simply represented as additional facts about the rule.

This approach to the representation of exceptions is used in Reason-Based Logic (chapter 2). We discussed the structure of rules and several types of facts concerning rules. Rules have a condition and a conclusion:

rule(*condition*, *conclusion*)

Rules can be valid, applicable and excluded, and can apply:

Valid(rule(*condition*, *conclusion*))
Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)
Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)
Applies(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)

The general properties of rules are defined by the relations that hold between these (and other) types of facts. The properties of rules (or classes of rules) are specified in the logical language. In contrast with the previous approach using rule identifiers, in this approach it is made explicit what is meant by 'rule' and by 'exception'.[17]

4.2 Exceptions and nonmonotonicity

Attempts to deal with nonmonotonic reasoning have become a vast field of research. Here we focus on rules with exceptions, and discuss three approaches.[18] The first is based on maxiconsistent sets. The second uses default rules. The third uses counterarguments.

Maxiconsistent sets

The first approach is based on maxiconsistent sets, as for instance used in Poole's Logical Framework for Default Reasoning (Poole, 1988).[19] We start by giving a brief overview of the definitions we need.

Poole's framework uses the language of First-Order Predicate Logic; for simplicity we use that of Propositional Logic. Assumptions are encoded in a *theory*, defined as a pair of sets (F, Δ), where F and Δ are both sets of sentences. F represents the strict assumptions, Δ the default assumptions. An *extension* of (F, Δ) is the set of consequences of a maximal *scenario*, where a scenario is a consistent set F ∪ D with D a subset of Δ.

A theory (F, Δ) has one or several extensions. Just as in Reiter's Default Logic (Reiter, 1980, 1987), a *credulous* and a *skeptical* consequence notion can be defined.

Maxiconsistent sets can be used to deal with reasoning with exceptions. For simplicity, we use a simple representation of rules here. A rule and its exception clause is represented as the following material conditional:

(1) *Condition* ∧ ¬Exception(*identifier*) → *Conclusion*.

(Below the number (1) is used to refer to this material conditional.) Rules are elements of the strict assumptions F, and different rules should have different identifiers. As default assumptions, we have that rules have no exceptions. Formally this is achieved by including assumptions of the following form:

¬Exception(*identifier*)

[17] In Reason-Based Logic, there are even different ways of representing exceptions, as discussed in chapter 3, section 5.

[18] Many of the observations in this section have been made before (cf., e.g., Prakken, 1993a). See note 2.

[19] We stress that we only use the maxiconsistent sets of Poole's framework (1988) here, and *not* his way to deal with rules, described in the same paper.

in the default information Δ, for all identifiers *identifier* corresponding to a rule occurring in F. In the following, if F = {Ass_1, Ass_2, ..., Ass_n}, Δ is as above, and $Conc_1$, $Conc_2$, ..., and $Conc_m$ are elements of all extensions of the theory (F, Δ), we write:

$Ass_1, Ass_2, ..., Ass_n \mid\!\sim Conc_1, Conc_2, ..., Conc_m$

Exceptions can now be represented by an exception rule in the strict information, as follows:

(2) *Exception* → Exception(*identifier*).

If there is no exception, the default assumption that there is no exception does not lead to a contradiction, so we have

Condition, (1), (2) $\mid\!\sim$ *Conclusion*, ¬Exception(*identifier*)

In the case of an exception, the exception rule (2) gives the following:

Condition, *Exception*, (1), (2) $\mid\!\sim$ Exception(*identifier*)

Since ¬Exception(*identifier*) does not follow, the conclusion of the rule does not follow, in other words:

Condition, *Exception*, (1), (2) $\mid\!\not\sim$ *Conclusion*

Since this corresponds to the two intuitive requirements STANDARD CASE and EXCEPTION CASE discussed in section 2.2, everything seems to work out fine.

However, a problem arises if there are exceptions to the exception rule itself. Exceptions to exceptions are a common phenomenon. In such a case the conclusion of the rule should follow in spite of the exception. We add a third intuitive requirement

EXCEPTION-TO-EXCEPTION CASE

If there is a rule the condition of which holds, there is an exception to the rule, and there is an exception to the exception, then the rule’s conclusion follows.

In order to meet this requirement, we need to represent exceptions to the exception rule. Therefore the exception rule (2) above is replaced by the following rule, that can have exceptions:

(3) *Exception* ∧ ¬Exception(*identifier2*) → Exception(*identifier*).

Furthermore we have an exception to the exception:

(4) *Exception_to_exception* → Exception(*identifier2*).

Since there is only one extension containing ¬Exception(*identifier*) and Exception(*identifier2*), we obtain the correct behavior in case of an exception to an exception:

Condition, *Exception*, *Exception_to_exception*, (1), (3), (4) |~ *Conclusion*

Unexpectedly, we have lost the correct behavior in the exception case: the theory (F, Δ) with F = {*Condition*, *Exception*, (1), (3), (4)} and Δ as above has two extensions. One extension contains both Exception(*identifier*) and ¬Exception(*identifier2*), as desired. The other contains ¬Exception(*identifier*), and *Conclusion*, but remains silent about whether the exception rule is applicable: it contains neither ¬Exception(*identifier2*) nor Exception(*identifier2*). It should be noted that in an extension containing ¬Exception(*identifier*) the inclusion of ¬Exception(*identifier2*) is blocked since that would give an inconsistency with (3). The exception rule (3) just demands that an extension that contains *Exception*, can only contain one of the sentences ¬Exception(*identifier*) and ¬Exception(*identifier2*). This demand is met in both extensions.

What is wrong is that the second extension does not contain the fact that there is no exception to the rule with identifier *identifier2*, only in order to maintain consistency. The first extension contains the fact that there is an exception to the rule *identifier* because there is an exception. Intuitively, we want that the application of a rule can only be blocked by explicit information in the extension.

Default rules

The second approach that we discuss uses Reiter's (1980, 1987) default rules. We use the definitions discussed in section 3.1. Default rules have, apart from a condition and a conclusion, a justification. This justification is used to block the application of a rule in case it follows that there is an exception. We do not have to assume by default that there is no exception to a rule, as in the previous approach.

Rules are represented as default rules as follows:

(5) *Condition* : ¬Exception(*identifier*) / *Conclusion*

Again it is assumed that different rules have different identifiers. An exception rule is represented as:

(6) *Exception* : ¬Exception(*identifier2*) / Exception(*identifier*)

An exception-to-exception rule is represented as:

(7) *Exception_to_exception* : ¬Exception(*identifier3*) / Exception(*identifier2*)

It turns out that the representation of rules and exceptions in this way meets the requirements, including that of the exception-to-exception case. To see the difference with the maxiconsistent set approach, we look what happens in the exception case that was problematic there.

We start with the default theory (F, Δ), where F = {*Condition*, *Exception*} and Δ = {(5), (6), (7)}. We propose two sets of sentences E and E* as guesses for extensions. They correspond to the two extensions in the maxiconsistent set approach:

E = Th(F ∪ {Exception(*identifier*)})[20]
E* = Th(F ∪ {¬Exception(*identifier2*), *Conclusion*})

The set E is indeed an extension, since we have:

E_0 = F, and
E_1 = Th(F ∪ {Exception(*identifier*)}) = E, and
$E_i = E_1$, for all $i > 1$.

and therefore $E = \bigcup_i E_i$, as required. But the set E* is not an extension. We have:

E^*_0 = F, and
E^*_1 = Th(F ∪ {*Conclusion*}), and
$E^*_i = E^*_1$, for all $i > 1$.

As a result $\bigcup_i E^*_i = E^*_1$, which is a proper subset of E*. Since no information in the assumptions supports that there is no exception to the rule *identifier2*, the sentence ¬Exception(*identifier2*) cannot be an element of an extension.

We conclude that this approach using default rules can adequately deal with the three requirements for reasoning with rules with exceptions.

Counterarguments

The third approach that we discuss uses counterarguments. We base the discussion here on Pollock's Theory of Defeasible Reasoning (1987-1995). We start by giving a description of some of his definitions, adapted to suit our needs.

An *argumentation theory* is a pair of sets (*Args*, *Defs*), such that *Defs* is a set of pairs of elements of *Args*. The elements of *Args* are called *arguments*, the elements of *Defs defeaters*. If (α, β) is an element of *Defs*, the argument α is said to *defeat* the argument β. Pollock then defines *levels*, as follows:

[20] Here Th(S) denotes the *deductive closure* of S, i.e., the set of all deductive consequences of S.

- All arguments are *in at level 0*.
- An argument is *in at level n + 1* if and only if it is in at level 0 and it is not defeated by any argument that is in at level n.

An argument is *ultimately undefeated* if and only if there is a level such that it is in at that level and at all higher levels. An argument is *ultimately defeated* if and only if there is a level such that it is out at all higher levels. An argument is *provisionally defeated* if and only if it is neither ultimately undefeated nor ultimately defeated.

Pollock's Theory of Defeasible Reasoning can be used to represent reasoning with rules with exceptions as follows.[21] We define a *theory of reasoning* as a pair of sets (*Facts*, *Rules*), where *Facts* are elements of some language L and Rules have the form *Condition* → *Conclusion*, where *Condition* and *Conclusion* are elements of the language. It is assumed that the language L contains identifiers for the rules in Rules, and has a predicate to represent exceptions. For instance, the fact that there is an exception to the rule *Condition* → *Conclusion* with identifier *id*, might be expressed as follows:

Exception(*id*)

For a theory of reasoning (*Facts*, *Rules*), we can define an argumentation theory (*Args*, *Defs*), as follows. The set *Args* consists of all facts and all loop-free chains of rules starting from the facts. The set *Defs* consists of pairs of arguments (α, β), such that the argument α ends with Exception(*id*), where *id* is the identifier of a rule in the argument β.

As an example, we assume that the set Rules consists of the following three rules, with identifiers *id1*, *id2* and *id3*, respectively:

Condition → *Conclusion*
Exception → Exception(*id1*)
Exception_to_exception → Exception(*id2*)

We discuss what happens in the standard, the exception and the exception-to-exception case. In the standard case, the set of facts only contains *Condition*. In that case, the only arguments are *Condition* and *Condition* → *Conclusion*, and there is no defeater. As a result, both arguments are in at all levels, and are ultimately undefeated.

In the exception case, the set of facts consists of *Condition* and *Exception*. There are two new arguments, namely *Exception* and *Exception* → Exception(*id1*). Now there is one defeater, namely (*Exception* → Exception(*id1*), *Condition* → *Conclusion*). We have:

[21] Here we do not follow Pollock.

The arguments *Condition*, *Condition* → *Conclusion*, *Exception* and *Exception* → Exception(*id1*) are in at level 0.
The arguments *Condition*, *Exception* and *Exception* → Exception(*id1*) are in at level 1, and at all higher levels.

So, the arguments *Condition*, *Exception* and *Exception* → Exception(*id1*) are ultimately undefeated, and the argument *Condition* → *Conclusion* is ultimately defeated. The latter argument is of course defeated by the argument *Exception* → Exception(*id1*).

In the exception-to-exception case, the set of facts consists of *Condition*, *Exception* and *Exception_to_exception*. The new arguments are *Exception_to_exception* and *Exception_to_exception* → Exception(*id2*). The new defeater is (*Exception_to_exception* → Exception(*id2*), *Exception* → Exception(*id1*)). We have:

The arguments *Condition*, *Condition* → *Conclusion*, *Exception*, *Exception* → Exception(*id1*), *Exception_to_exception* and *Exception_to_exception* → Exception(*id2*) are in at level 0.
The arguments *Condition*, *Exception*, *Exception_to_exception* and *Exception_to_exception* → Exception(*id2*) are in at level 1.
The arguments *Condition*, *Condition* → *Conclusion*, *Exception*, *Exception_to_exception* and *Exception_to_exception* → Exception(*id2*) are in at level 2, and at all higher levels.

So, all arguments are ultimately undefeated, except the argument *Exception* → Exception(*id1*), that is ultimately defeated. The latter argument is defeated by the argument *Exception_to_exception* → Exception(*id2*).

5 Rule conflicts

In this section, we discuss approaches to dealing with rule conflicts. We do this in two parts. First, we discuss different approaches to the representation of conflict resolving information. Second, we discuss approaches to dealing with conflicts and consistency maintenance.

5.1 Representing conflict resolving information

We start with a discussion of approaches to the representation of conflict resolving information. We distinguish three types of conflicts: conflicts of pairs of rules, bipolar multiple conflicts, and general multiple conflicts. For each type of conflict, we discuss a corresponding type of conflict resolving information: rule priorities, weighing, and general conflict resolution, respectively.

Conflicts of pairs of rules and rule priorities

The simplest, and most common, type of rule conflict is the conflict of two rules: there are two rules with opposite conclusions and the conditions of both rules are satisfied.

If there is a conflict of a pair of rules, often one of the rules prevails over the other. We have seen several examples in chapter 3, section 6.1. As a result of such priority information, the conflict is resolved. The prevailing rule leads to its conclusion, while the other rule does not. Clearly, the conflict of rules leads to a special type of exception to the non-prevailing rule. As a result, rule priorities can be represented using the techniques already discussed in section 4.1 on representing exceptions.

Assume that we have two rules with incompatible conclusions represented as the following two material conditionals:

$$\mathit{Condition}_1 \wedge \neg\mathrm{Exception}(\mathit{identifier}_1) \rightarrow \mathit{Conclusion}$$
$$\mathit{Condition}_2 \wedge \neg\mathrm{Exception}(\mathit{identifier}_2) \rightarrow \neg\mathit{Conclusion}$$

Assume moreover that the first prevails over the other. This priority information can now be represented as follows:

$$\mathit{Condition}_1 \wedge \neg\mathrm{Exception}(\mathit{identifier}_1) \rightarrow \mathrm{Exception}(\mathit{identifier}_2)$$

Abbreviating $\mathit{Condition}_i \wedge \neg\mathrm{Exception}(\mathit{identifier}_i)$ as $\mathrm{Applicable}(\mathit{identifier}_i)$ (for i = 1 or 2), we obtain the following sentence:

$$\mathrm{Applicable}(\mathit{identifier}_1) \rightarrow \mathrm{Exception}(\mathit{identifier}_2)$$

It may be tempting to represent the priority information as the following sentence:

$$\mathrm{Applicable}(\mathit{identifier}_1) \rightarrow \neg\mathrm{Applicable}(\mathit{identifier}_2)$$

However, this is an incorrect representation, since this sentence is symmetric in the two rules, as its equivalent

$$\neg\mathrm{Applicable}(\mathit{identifier}_1) \vee \neg\mathrm{Applicable}(\mathit{identifier}_2)$$

clearly shows.

Bipolar multiple conflicts and weighing

The second type of rule conflict that we discuss are bipolar multiple conflicts: two groups of rules have equal conclusions in each group, but incompatible conclusions across the groups, while the conditions of the rules are satisfied.

For instance, the following material conditionals represent a bipolar conflict of a group of n rules and a group of m rules (where n and m are natural numbers):

$Condition_{11} \wedge \neg$Exception($identifier_{11}$) $\rightarrow$ *Conclusion*
...
$Condition_{1n} \wedge \neg$Exception($identifier_{1n}$) $\rightarrow$ *Conclusion*
$Condition_{21} \wedge \neg$Exception($identifier_{21}$) $\rightarrow \neg$*Conclusion*
...
$Condition_{2m} \wedge \neg$Exception($identifier_{2m}$) $\rightarrow \neg$*Conclusion*

We have seen examples in which such a conflict cannot be resolved by priority information on pairs of rules, but by priority information on *groups* of rules (chapter 2, section 1.3; chapter 3, section 4).

The priority technique used for pairwise conflicts can be extended to the case of bipolar multiple conflicts. For instance, if the first group of n rules above prevails over the second group of m rules, this can be represented as follows:

Applicable($identifier_{11}$) $\wedge$ ... $\wedge$ Applicable($identifier_{1n}$) $\rightarrow$ Exception($identifier_{21}$) $\wedge$
... $\wedge$ Exception($identifier_{2m}$)

In Reason-Based Logic (chapter 2), a representation similar to this one is possible. However, Reason-Based Logic provides a second way of representation, using the weighing of reasons. The priority of the first group of rules over the second is represented as the fact that the reasons that result from the first group of rules outweigh the reasons from the second group:

Outweighs({$condition_{11}$, ..., $condition_{1n}$},
{$condition_{21}$, ..., $condition_{2m}$},
conclusion)

The two techniques seem to lead to similar results. However, there is a technical difference. The two expressions representing conflict resolving information are not equivalent, because the weighing expression only helps to resolve the conflict if there is no other rule with conclusion ¬*conclusion* (cf. the relations between facts described in chapter 2, section 5), while the generalized priority expression helps to resolve the conflict also in that case. The use of the weighing expression reflects the intuition that the bipolar multiple conflict should only be resolved if all rules of the losing side, i.e., those with conclusion ¬*conclusion*, have been considered. In the more familiar terminology of reasons, the weighing information only should have effect if all counterreasons have been considered.

As a result, the explicit representation of the weighing of reasons as in Reason-Based Logic seems to be closer to the examples of accrual of reasons, that led to the distinction of bipolar rule conflicts.

General multiple conflicts and general conflict resolution

As a third type of rule conflict, we discuss general rule conflicts: there is a group of rules with incompatible conclusions, the conditions of which are satisfied. For instance, we might have:

$Condition_1 \wedge \neg\text{Exception}(identifier_1) \rightarrow Conclusion_1$
...
$Condition_n \wedge \neg\text{Exception}(identifier_n) \rightarrow Conclusion_n$
$\neg(Conclusion_1 \wedge ... \wedge Conclusion_n)$

We have seen two special cases of resolutions of such a general rule conflict:

1. One of the rules might prevail over another.
2. A subgroup of rules might prevail over another subgroup of rules with incompatible conclusion.

The most general type of conflict resolution would require the representation of the prevalence of *any* subgroup over *any* other subgroup, formally:

$\text{Prevails}(\{identifier_{11}, ..., identifier_{1n}\},$
$\{identifier_{21}, ..., identifier_{2m}\})$

We do not know a formalism in which this is explicitly done, although it is a natural generalization of the two discussed representation techniques, i.e., using exceptions and using weighing, to the case of general multiple conflicts.

5.2 Conflicts and consistency maintenance

Since there is not always sufficient information to resolve rule conflicts, many techniques have been proposed to prevent the unwanted effects of contradiction by means of consistency maintenance. Here we discuss three such techniques. We start with Reiter's normal and semi-normal default rules (Reiter, 1980, 1987), then we discuss Vreeswijk's use of conclusive force (Vreeswijk, 1991, 1993), and we finish with Pollock's collective defeat (Pollock, 1987).

Normal and semi-normal default rules

The first approach to consistency maintenance in case of rule conflicts that we discuss are the normal and semi-normal default rules of Reiter's Default Logic (Reiter, 1980, 1987). In section 4.2, we already discussed how default rules (Reiter, 1980, 1987) can be used to represent rules with exceptions. There, a rule was represented as a default of the following form:

Condition : ¬Exception(*identifier*) / *Conclusion*

If two default rules of this form are in conflict, there is no extension. An example is the theory (F, Δ) defined as follows:

F = {*Condition*$_1$, *Condition*$_2$}
Δ = {*Condition*$_1$: ¬Exception(*identifier*$_1$) / *Conclusion*,
Condition$_2$: ¬Exception(*identifier*$_2$) / ¬*Conclusion*}

As a result, using the skeptical consequence relation of Default Logic, everything follows from such a theory. This behavior resembles the behavior of an inconsistency in classical logic.

There is another type of default rule that can never give rise to this behavior: default rules of this type are called normal default rules, and have the form

Condition : *Conclusion* / Conclusion.

Informally, a default rule leads to its conclusion if its condition is satisfied, unless that would lead to an inconsistency. Normal defaults have the nice formal property that a theory that only contains normal default rules always has an extension.

Reiter (1980, 1987) claimed that normal default rules were sufficient in practice. However, as was already noted by Reiter and Criscuolo (1981, 1987), normal default rules are not always sufficient. We saw above that non-normal default rules are needed to represent rules with exceptions.

In order to catch the benefits of both, a combined form can be used, as follows:

Condition : ¬Exception(*identifier*), *Conclusion* / *Conclusion*

Default rules that have their conclusion as one of their justifications are called semi-normal. Informally, a default rule of this form leads to its conclusion if its condition is satisfied, unless Exception(*identifier*) or ¬*Conclusion* would also follow.

Two conflicting rules will now give rise to two extensions. For instance, the theory (F, Δ) with

F = {*Condition*$_1$, *Condition*$_2$}
Δ = {*Condition*$_1$: ¬Exception(*identifier*$_1$), *Conclusion* / *Conclusion*,
Condition$_2$: ¬Exception(*identifier*$_2$), ¬*Conclusion* / ¬*Conclusion*}

has two extensions E_1 and E_2:

E_1 = Th({*Condition*$_1$, *Condition*$_2$, *Conclusion*})
E_2 = Th({*Condition*$_1$, *Condition*$_2$, ¬*Conclusion*})

In each extension, only one of the rules has led to its conclusion. Intuitively, the two extensions can arise because there are two orders in which the defaults can be used: first drawing the conclusion of rule $identifier_1$ blocks using rule $identifier_2$, while first drawing the conclusion of $identifier_2$ blocks using rule $identifier_1$.

However, a theory with only semi-normal default rules does not always have an extension, as the theory (F, Δ) with

$F = \{Condition_1, Condition_2, Condition_3\}$
$\Delta = \{Condition_1 : \neg\text{Exception}(id_1), \text{Exception}(id_2) / \text{Exception}(id_2),$
$Condition_2 : \neg\text{Exception}(id_2), \text{Exception}(id_3) / \text{Exception}(id_3),$
$Condition_3 : \neg\text{Exception}(id_3), \text{Exception}(id_1) / \text{Exception}(id_1)\}$

shows.

Conclusive force

The second approach to consistency maintenance in case of conflicts that we discuss is Vreeswijk's use of the conclusive force of arguments. We give a simplified overview of some definitions of Vreeswijk's (1991, 1993) Abstract Argumentation Systems.

Vreeswijk starts with the definition of an *argumentation system* as a triple (*Language*, *Rules*, <). Here *Language* is any set containing a special element ⊥, denoting contradiction. This set is called the *language* of the argumentation system. The set *Rules* is a set of rules, that have the form $Condition_1, ..., Condition_n \rightarrow Conclusion$. The *conclusive force relation* < is a strict order on arguments, that are tree-like chains of rules.

He proceeds with the definition of defeasible entailment and extensions, which uses the notion of conflict. A set of arguments *Arguments* is *in conflict with* an argument *Argument* (relative to a set $Assumptions \subseteq Language$), if *Argument* and elements of *Arguments* are parts of a larger argument with conclusion ⊥ and with premises in the set *Assumptions*. A relation |~ between sets of sentences of the language and arguments is called a *defeasible entailment relation* if the following holds for all sets $Facts \subseteq Language$ and arguments *Argument*:

Assumptions |~ *Argument* if and only if one of the following holds:

1. *Argument* is an element of *Assumptions*
2. *Argument* has the form $Argument_1, ..., Argument_2 \rightarrow Conclusion$, and for every set of arguments *Arguments*, such that *Assumptions* |~ *Argument'* for all elements *Argument*' of *Arguments*, we have:
   If *Arguments* is in conflict with *Argument* (relative to *Assumptions*), then there is a *Argument'* in *Arguments*, such that *Argument*' < *Argument*.

(This is not a definition of the relation |~ by recursion on arguments, since |~ appears on both sides of the 'if and only if'. Such a definition would be unexpected since $\mid\!\sim$ is nonmonotonic.) An *extension* of a set *Assumptions* is then defined as a set of arguments *Arguments*, such that *Arguments* = {*Argument* | *Assumptions* $\mid\!\sim$ *Argument*}.

How can Vreeswijk's formalism be used to maintain consistency in case of rule conflicts? In Vreeswijk's formalism, conflicts of rules occur as conflicts of the final steps of arguments. Informally, Vreeswijk's definition has the result that such conflicts between arguments are resolved by 'throwing away' one argument that is involved in the conflict.

To choose an argument, the conclusive force relation is used: an argument cannot be thrown away if it is stronger than any of the other arguments involved in the conflict. Since there can still remain more than one argument that can be chosen, multiple extensions can arise.

Collective defeat

As a third approach to consistency maintenance in case of rule conflicts, we mention Pollock's collective defeat (Pollock, 1987).

He proposes to withhold from drawing a conclusion in case there is an unresolved conflict of rules. He achieves this by considering all arguments with conflicting last steps as defeated in case of an unresolved conflict: the arguments are *collectively defeated*.

In Reason-Based Logic (chapter 2), there is a variant of collective defeat: if there are conflicting reasons, but there is no weighing information available, no conclusion follows. As a result, while Pollock's collective defeat can maintain consistency for general multiple rule conflicts, Reason-Based Logic uses a form of collective defeat in the specific case of bipolar multiple conflicts.

6 Reasoning about rules

Below, we discuss three approaches to dealing with reasoning about rules. The first is to treat rules as special sentences. The second is to use rule identifiers. The third is to treat rules as special objects.

6.1 Rules as special sentences

The first approach to dealing with reasoning about rules is to treat rules as special sentences, as in conditional logics (e.g., Nute, 1980, 1994; Delgrande, 1988). In conditional logics, the properties of a special connective are specified on the meta-level. We already discussed the conditional logic approach in section 3.2 on relevance. There, we mentioned conditional logics because they make it possible to take the requirement of relevance into account. But conditional logics also are regarded as logics that can treat reasoning with rules.

For instance, assume we want to represent a transitive type of rule. Then one of the defining properties of the rule-representing conditional would be:

$A > B, B > C \mid\!\sim A > C$

This rule of inference makes it possible to derive the conditional $A > C$ from the conditionals $A > B$ and $B > C$.

If we wish to represent a type of rule that is not transitive, one of the defining properties of the conditional representing the rule type, might have the following weaker property:

$A > B, (A \wedge B) > C \mid\!\sim A > C$

By choosing the defining properties, we can specify different forms of reasoning about rules.

However, there are two limitations. The first limitation is that in this way it is impossible to distinguish classes of rules unless each class of rules is represented by a syntactically different conditional. As a result, properties of rules of different kinds, such as transitive and intransitive rules, can only be represented at the meta-level, and not, more flexibly, at the logical level.

The second limitation is that in conditional logics, it is impossible to represent facts about rules, other than that they are valid or invalid. As a result, although it is possible to represent the first type of reasoning about rules distinguished in section 2.4, i.e., reasoning with rules as conclusions, it is not possible to represent the second type, i.e., reasoning with facts about rules.

6.2 Rule identifiers

The second approach to dealing with reasoning about rules is an attempt to deal with these limitations, and is the technique of rule identifiers, already discussed as one of the techniques to represent exceptions in section 4.1.

This technique can be used in a more general way to deal with reasoning about rules. To represent exceptions to rules the rule identifiers were only used in the special purpose predicate Exception(*identifier*). However, rule identifiers can also be used as parameters for other predicates. For instance, the conclusion of a priority argument says that some rule prevails over another rule. If the identifiers of these rules are $identifier_1$ and $identifier_2$, this can be represented as follows:

Prevails($identifier_1$, $identifier_2$)

As a result, if this technique is used, it is possible to represent the second type of reasoning about rules distinguished in section 2.4, i.e., reasoning with facts about rules, but it is not clear how to represent the first type, i.e., reasoning with rules as conclusions. This limitation is the result of the fact that the approach is unclear about the status of rules and identifiers, as was already noted in section 4.1 on representing exceptions.

6.3 Rules as special objects

The latter brings us to the third approach to dealing with reasoning about rules, namely to consider rules as special objects, as in Reason-Based Logic (chapter 2). This approach was also discussed in section 3.3 on relevance and section 4.1 on representing exceptions.

As we will see, this approach can be regarded as an integration of the two other approaches, keeping the benefits of both. To recall, rules are treated as objects, that are represented as follows:

rule(*condition*, *conclusion*)

Transitivity of rules can now be represented as

Valid(rule(*a*, *b*)), Valid(rule(*b*, *c*)) |~ Valid(rule(*a*, *c*))

Transitivity of rules can be restricted to a certain class of rules by explicitly mentioning such a class, here named transitive_class:

Class(transitive_class, rule(*a*, *b*)), Class(transitive_class, rule(*b*, *c*)),
Valid(rule(*a*, *b*)), Valid(rule(*b*, *c*)) |~ Valid(rule(*a*, *c*))

Facts about rules are stated using the complete rule, instead of using only an identifier. The former is more expressive. For instance, whereas for the representation of an exception to a rule it is sufficient to use an identifier, as in

Excluded(*identifier*),

for the representation of the validity of a rule the complete rule, including its condition and conclusion are needed, as in the following sentence:

Valid(rule(*condition*, *conclusion*))

As explained in chapter 2, section 3.1, this approach requires a translation from sentences to terms. This translation is already necessary in order to draw a conclusion from a valid rule, as in ordinary rule application:

Valid(rule(*condition*, *conclusion*)), *Condition* |~ *Conclusion*

For details on the formal definition of such a translation, the reader is referred to chapter 2, section 4.3.

In this approach, the properties of rules can be specified both on the meta-level and on the level of the logical language. As a result, the meta-level can be used to specify the general properties of rules that are considered most basic, such as rule application and its relation to exceptions, while the level of the logical language can be used to specify the specific properties of specific rules in a specific case or domain.

Therefore, this approach can deal with both types of reasoning with rules that were distinguished in section 2.4, in contrast with the conditionals and identifiers approach.

Version November 12, 1999