Chapter 2

Reason-Based Logic: a semantics of rules and reasons

In this chapter, a formalism is developed that models rules and reasons. The formalism, called Reason-Based Logic, is a formal semantics of rules and reasons: Reason-Based Logic specifies the types of facts concerning rules and reasons that are relevant for the defeasibility of arguments, and makes the relations that must hold between these facts precise.[1] Examples of such facts are the fact that some rule applies, or that certain reasons outweigh other reasons. A crucial difference with other logical formalisms is that Reason-Based Logic provides a semantics in which such facts and their relations are made explicit.

The chapter begins with a motivation of the approach by means of examples (section 1). After a discussion of what is meant by a formal semantics (section 2), the formalism is introduced using the informal examples (section 3). Then a description of the formalism follows. First the types of facts concerning rules and reasons, as distinguished in Reason-Based Logic, are described (section 4), and second the relations between these types of facts (section 5). Third we define which conclusions follow from given premises (section 6).

1 Rules and reasons by example

In the previous chapter, we introduced argumentation by concentrating on the arguments that can justify a conclusion, and their defeasibility. In this chapter, we focus on rules and reasons. Both are fundamental for argumentation: *rules* give rise to the *reasons* that are used in arguments to support a conclusion. We start with a

---

[1] Hage initiated the development of Reason-Based Logic; it was continued in close cooperation with Verheij. Hage (1991) describes a theory of rational belief, called Reason Based Reasoning, that already contains the basic informal ideas of Reason-Based Logic. Verheij (1994) describes a limited version of Reason-Based Logic to get the formalism right. Hage and Verheij (1994) describe the first full version of Reason-Based Logic that is also formally satisfactory. The description of Reason-Based Logic in this chapter as a specification of types of facts concerning rules and reasons and the relations between these facts is related to that of Verheij (1995e). See also note 14.

number of informal examples. Issues related to the defeasibility of arguments (introduced in chapter 1, section 4) are examined in detail.

1.1 Rules and reasons

> Mary and John are planning to have a picnic on Sunday. The evening before, they watch the weather report on television. According to the weather report, it is going to rain the whole day. Mary and John are disappointed.

Mary and John's disappointment is the result of the following argument:

> According to the weather report, it will rain all day.
> So, it will rain all day.

Because of this argument John and Mary conclude that it will rain all day. Their conclusion is in this argument supported by the prediction in the weather report. 'According to the weather report, it will rain all day' is a reason for 'It will rain all day'.

John and Mary would have made a similar argument if the prediction in the weather report had been different. If the prediction had been that it would be a sunny day, John and Mary would have concluded that it will be sunny because of the weather report.

So, reasons do not arise individually, but follow a pattern. The prediction of the weather report gives rise to a reason, whatever that prediction is, in the following pattern:

> According to the weather report, it will *be weather type so-and-so*.
> So, it will *be weather type so-and-so*.

Each instance of this argument scheme can be an argument that supports the conclusion that it will be some type of weather. Moreover, each instance can be a step in a larger argument. The relation between a reason and a conclusion as expressed by such an argument scheme is what we call a *rule*.[2] If an instance of the scheme can actually be used as part of an argument that supports its conclusion (for instance, when its condition holds) we say that the rule *applies*.

Not all rules give rise to argument schemes that lead to acceptable arguments. We consider a rule to be *valid* if it is generally accepted (in some reasoning community) that the application of the rule can give rise to an argument that supports its conclusion.[3]

---

[2] Rules in Reason-Based Logic correspond to warrants in Toulmin's (1958) argument scheme.

[3] This is in contrast with the *legal validity* of a rule, which requires that the rule is obtained by a legal procedure, such as when the rule is made by the legislator, and approved by the parliament.

1.2 Exclusionary reasons

> After watching the weather report, John is disappointed. He would not like to have a picnic if it is going to rain all day. Mary smiles, and says that he does not have to worry, because the weather report on national television is not good at predicting the weather in their district, due to the peculiar local circumstances. Therefore, there is no reason to conclude that it will rain all day.

The story illustrates the defeasibility of arguments, introduced in the previous chapter. In chapter 1, the first example of the defeat of an argument was an exception to a rule (chapter 1, section 4.1): in exceptional circumstances the conclusion of a rule does not follow, even though its condition holds.

In the story about John and Mary, we again encounter an example of an exception: the weather report on national television is not good at predicting the local weather. Therefore, the fact that, according to the weather report, it will rain all day is *not* a reason that it will rain all day in this district. The rule underlying the argument scheme

> According to the weather report, it will *be weather type so-and-so*.
> So, it will *be weather type so-and-so*.

is not applicable, even though its condition is satisfied by the fact that, according to the weather report, it will rain all day. We say that 'The weather report on national television is not good at predicting the local weather' is an *exclusionary reason* to the applicability of the rule.[4] In case there is no exclusionary reason to the applicability of a rule, the rule is applicable. We will later see that even an applicable rule does not always apply, although it normally does (section 1.4).

1.3 Weighing reasons

> That Saturday evening, John's father pays a visit, and the plan to have a picnic is discussed. He agrees that the weather report on television is not good at predicting the local weather, but says that he nevertheless thinks that it will rain on Sunday. Because John's father has been a farmer for more than twenty years, John and Mary take his opinion seriously. They go to bed disappointedly. The next morning Mary looks out the window and sees that the sky is completely cloudless. She happily tells John that it might not rain after all.

---

[4] Our use of the term 'exclusionary reason' is closely related to Raz's (1990, p. 35ff.). Raz focuses on reasons for acting, and he defines an exclusionary reason as a reason not to act for some other reason. Our exclusionary reasons are reasons that make a rule inapplicable, even in case its condition is satisfied.

At this point in our story, John and Mary can make two arguments, one that it will rain:

> John’s father thinks that it will rain.
> So, it will rain.

and the other that it will not rain:

> The sky is completely cloudless.
> So, it will not rain.

This is an instance of conflicting arguments (see chapter 1, section 4.2). Because there is a reason that it will rain, and also a reason that it will not rain, John and Mary can currently not draw a conclusion.

> At breakfast, John says he is at a loss, and does not know what to think about the weather. He still takes his father’s opinion seriously, but agrees with Mary that the weather looks very good. After some discussion, John and Mary decide that what they see with their own eyes provides the stronger reason, and they conclude it will not rain.

In the story, John and Mary have *weighed* the conflicting reasons.[5] Since John and Mary consider the second reason the strongest, the argument

> The sky is completely cloudless.
> So, it will not rain.

justifies its conclusion, while the argument

> John’s father thinks that it will rain.
> So, it will rain.

does not, and John and Mary conclude that it will not rain.

Weighing can involve several reasons for and against a conclusion. If, for instance, the prediction of the national weather report had been good at predicting the local weather, and therefore the rule based on the prediction was not excluded, there would have been an additional reason that it will rain. In that situation, the reasons would again have to be weighed. John and Mary might still decide that what they see with their own eyes gives a reason that is strong enough to outweigh both opposing reasons, but they might also change their opinion and decide that the reasons provided by the weather report and the opinion of John’s father together are stronger than the cloudless sky alone.

---

[5] Cf. Naess (1978), p. 100ff.

1.4 Reasons concerning the application of a rule

> After preparing the food, John and Mary drive off to their favorite picnic site. They turn their local radio station on, and at ten o'clock the weather report brings bad news: after a nice start of the day, it will begin raining before noon. John and Mary know that, in contrast with the national weather report on television, this local weather report provides a strong reason that it will rain. Nevertheless, they refuse to take it into account, against better judgment.

John and Mary's seemingly irrational behavior has a reason: otherwise, they would have to conclude that it will rain, and they would certainly not enjoy their trip any longer. As before, John and Mary consider the rule underlying the argument scheme below to be valid:

> According to the weather report, it will *be weather type so-and-so*.
> So, it will *be weather type so-and-so*.

In the case of the report on television, this rule was excluded, because the national report is not good at predicting the local weather. This exclusionary reason does not hold for the local weather report on the radio. Nevertheless, John and Mary do not take the reason that it will rain into account. In other words, they do not apply the rule.

Nevertheless, they have a reason for applying the rule since the rule is applicable: the condition of the rule is satisfied, and the rule is not excluded. They also have a reason against applying the rule since if they would apply it they would certainly not enjoy their trip any longer. Their arguments are the following:

> The rule's condition is satisfied.
> So, the rule applies.

and

> The trip will certainly not be enjoyable any longer if the rule is applied.
> So, the rule does not apply.

Again there is a conflict of reasons, and the reasons have to be weighed. In this case, John and Mary consider the reason not to apply the rule to be the strongest.

The seemingly irrational behavior of Mary and John shows an important characteristic of rule application: it is an act, and there can be reasons for and against performing the act. Their behavior is only *seemingly* irrational: John and Mary do have a reason not to apply the rule.[6]

---

[6] In chapter 3, section 5.2, another example of reasons against the application of a rule is discussed taken from the field of law.

And, for those who may wonder, Mary and John's behavior *did* have the right result: the weather stayed well during their picnic, and they had a nice afternoon. Only when they got back in their car, did it begin to rain heavily.

1.5 Overview

In the remainder of this chapter we will forget about the actual practice of argumentation (to which we will return in chapter 5), and focus on the rules and reasons on which argumentation is based. The resulting model of rules and reasons can be used to analyze argumentation. As we hope our examples have shown, such a model is bound to be rather complicated.

In the examples, we made the following points about rules and reasons:

- Reasons for a conclusion do not arise individually, but follow a pattern represented by a valid rule.
- By the application of a rule, a reason arises that supports a conclusion in an argument.
- A rule can be excluded if there is an exclusionary reason. An excluded rule is not applicable, even if its condition is satisfied.
- In case of conflicting reasons, whether a conclusion follows depends on how the reasons pro and con are weighed. The outcome of the weighing can change if new reasons arise.
- The application of a rule is an act. There can be reasons for and against performing the act. If a rule is applicable, the fact that makes it applicable is a reason to apply the rule.

The remainder of this chapter is devoted to the elaboration of these points and to the development of a formalism called Reason-Based Logic that is based on them.

2 Semantics

In the previous section, we have informally introduced our view on the role of rules and reasons in argumentation with defeasible arguments, by means of examples. This view is at the core of Reason-Based Logic. Using these examples, we develop the formalism Reason-Based Logic in the subsequent sections, in accordance with our method of research (chapter 1, section 7). Reason-Based Logic can be regarded as a formal semantics of rules and reasons. In this section, we explain what we mean by this.

We introduce some convenient terminology. In the world there are *facts*. For instance, it can be a fact that the earth is round and that there is an oak tree in the park. Facts can be expressed by *sentences* in some language. For instance, the fact that the earth is round can be expressed in English as 'The earth is round' and in Dutch as 'De aarde is rond'. Not all sentences express facts. For instance, if it is a fact that the earth is round, then ‘The earth is flat’ does not express a fact. A sentence that expresses a fact is *true*. We call a part of the world that is expressed by a sentence, whether it is true or not, a *state of affairs*. Both sentences ‘The earth is round’ and ‘The earth is flat’ express states of affairs, but only one of them can express a fact.

Not all facts deal with physical objects, such as the earth or oak trees. In this chapter, for instance, we are particularly interested in objects related to argumentation, such as rules and reasons. It can be a fact that one reason outweighs another reason, or that there is an exception to a rule.

Facts are not isolated, but stand in relation to each other. An example is the combination of facts by conjunction: it is a fact that the earth is round and it is a fact that there is an oak tree in the park if and only if it is a fact that the earth is round and there is an oak tree in the park. If we look at the corresponding sentences, we obtain the following:

> The sentence ‘The earth is round’ is true and the sentence ‘There is an oak tree in the park’ is true if and only if the sentence ‘The earth is round and there is an oak tree in the park’ is true.

We give another example, that is related to argumentation: if it is a fact that Mary’s argument justifies its conclusion, then it is also a fact that there are applying rules that give rise to the steps in Mary’s argument.

We call a specification of the types of facts in some domain and the relations that hold between these facts a *semantics* of that domain.[7] Since facts can be expressed as sentences of some language, the types of facts in a domain are specified by defining an appropriate language. The relations that hold between facts are specified in terms of relations between the truth values of sentences.

A well-known example is the ‘domain of the logical connectives’, and its well-known Tarski semantics.[8] One of the types of facts in this domain is conjunction. In terms of sentences, ‘*S1* and *S2*’ expresses the conjunction of the facts expressed by ‘*S1*’ and ‘*S2*’. The relation that holds between facts combined by conjunction is, in terms of the corresponding sentences:

> ‘*S1*’ is true and ‘*S2*’ is true if and only if ‘*S1* and *S2*’ are true.

For the other logical connectives, similar relations hold.

In the domain of the logical connectives, the truth value of a sentence is determined by the truth values of its building blocks, such as in the example of

---

7 We use this terminology in analogy with that of the Tarski semantics in formal logic (see e.g., Davis, 1993, p. 34ff.). However, in style the semantics of rules and reasons discussed in this chapter differs, and is comparable to the representations of the commonsense world, as discussed by, e.g., Hayes (1985), Hobbs and Moore (1985), and Davis (1990).

8 See Haack (1978, p. 108ff.) for a philosophical account, or any introductory text on formal logic, e.g., Davis (1993, p. 34ff.), for a formal account.

conjunction. The logical connectives are said to be truth-functional. In other domains this is not always the case. For instance, the truth value of the sentence ‘John loves Mary’ depends on the truth value of the sentence ‘John hates Mary’. In a semantics of love and hate this relation has to be specified. The semantics might for instance state that ‘John loves Mary’ and ‘John hates Mary’ cannot both be true.

In this chapter, we describe a semantics of the domain of rules and reasons. Also in this domain the truth value of sentences is not solely determined by the truth value of their building blocks. An example of a relation between truth values of sentences in this domain is:

> If ‘The rule with conclusion *conclusion* and condition *condition* is excluded’ is true, then ‘The rule with conclusion *conclusion* and condition *condition* is valid’ is true.

Here *condition* and *conclusion* are variables that stand for the condition and conclusion of a rule. Informally, this relation between true sentences says that only valid rules can be excluded.

The above shows a second difference between the domain of rules and reasons and the domain of the logical connectives: there exists a semantics of the logical connectives that is generally agreed upon, namely the Tarski semantics. This semantics is so well-known that it seems to be the obviously right one, and even a ‘silly pedantic exercise’ (Davis, 1993, p. 34). In the domain of rules and reasons, however, this is not the case. There is no general agreement on the elementary concepts nor on their relations. This adds to the importance of our method of research: any attempt to describe a semantics of rules and reasons should be accompanied by informal examples (cf. chapter 1, section 7).

The validity of rules and the existence of reasons are the bottom line of our treatment of argumentation: our semantics of rules and reasons does not define which rules are valid, and which reasons exist. In our view, such facts can only be determined by means of empirical investigation: which reasons exist and which rules are valid in a given reasoning community is shown by the argumentation behavior of the reasoners in that community. Which rules are valid and which reasons exist is not determined by logic.

Just as with other empirically studied domains it cannot be expected that the empirical data lead to a unique and indisputable theory of rule validity. Moreover, it will often happen that new data gives rise to a revision of the theory of rule validity. To complicate matters further, rule validity can change with time and there is not always general agreement about rule validity in a community.[9] Therefore,

[9] Even in the mathematical community where argumentation takes the form of mathematical proof the ideas about rule validity can change and can be the subject of dispute. Examples are Brouwer’s constructivist view on mathematical proof, and more recently the dispute about the acceptability of computers as proof tools (cf. Stewart, 1996).

examples are always based on a theory of rules and reasons that is given beforehand as a set of premises.

To summarize, the goal of this chapter is twofold:

- We specify types of facts concerning rules and reasons by defining an appropriate language containing sentences that express these types of facts (section 4).
- We specify the relations that must hold between the types of facts, in terms of the relations between the truth values of sentences in this language (section 5).

After the formal description of the semantics of rules and reasons, we discuss which conclusions follow from given premises (section 6). In agreement with our method (chapter 1, section 7), we continue with an introduction of the formalism, by means of the examples from section 1.

3 Towards a formalization

In the examples of section 1, we have encountered several types of facts concerning rules and reasons. For instance, a rule can be valid, it can be applied, and reasons can be weighed. Reason-Based Logic is a formalism in which such facts can be formally represented, and that makes the relations between these facts precise. In this section, we use the informal examples of section 1 to introduce this formalism.

3.1 Rules and reasons

The conclusion of an argument is supported by reasons. For instance, in the argument

> According to the weather report, it will rain all day.
> So, it will rain all day.

‘According to the weather report, it will rain all day’ is a reason for ‘It will rain all day’.[10] As the language of Reason-Based Logic, we use the language of First-Order Predicate Logic.[11] A number of special function and predicate symbols are used to express the notions that are typical for Reason-Based Logic. The premise and the

---

[10] Actually, we should say that the *state of affairs* expressed by the sentence ‘According to the weather report, it will rain all day’ is a reason for *state of affairs* expressed by the sentence ‘It will rain all day’. (Cf. the difference between states of affairs and the sentences expressing them discussed in section 2.) For convenience, however, we will not use this extensive expression.

[11] For an introduction to First-Order Predicate Logic, see Van Dalen (1983) or Davis (1993).

conclusion of the argument above can be represented as Weather_report(rainy_day) and Rainy_day, respectively.

In the argument, Weather_report(rainy_day) is a reason for Rainy_day.[12] In Reason-Based Logic, a special predicate is used to express this fact:

Reason(weather_report(rainy_day),
 rainy_day)

This sentence expresses a state of affairs that some state of affairs is a reason for another. As a result, the sentence, expressing one state of affairs, contains references to other states of affairs. Here we encounter an important subtlety in the language of Reason-Based Logic: states of affairs are expressed by sentences of the language, and referred to by terms in other sentences. For instance, the state of affairs that, according to the weather report, it will rain all day, is expressed by the sentence

Weather_report(rainy_day)

and referred to by the term

weather_report(rainy_day)

in the sentence

Reason(weather_report(rainy_day),
 rainy_day).

As a result, in the language of Reason-Based Logic, there is a translation from sentences to terms. In order to distinguish between sentences expressing states of affairs and terms referring to them, a typographical convention is used: a string with an initial upper-case character is a sentence, and a string with an initial lower-case character a term (see section 4.3 for details).

We have discussed that reasons do not arise individually, but follow a pattern (section 1.1). The reason above instantiates the following reason scheme:

Reason(weather_report(*weather_type*),
 weather_type)

[12] It may seem sloppy that we use the same phrase '... is a reason for ...' in the sentence ' 'According to the weather report, it will rain all day' is a reason for 'It will rain all day' ' and in the sentence 'Weather_report(rainy_day) is a reason for Rainy_day'. However, no confusion can arise, since both 'According to the weather report, it will rain all day' and Weather_report(rainy_day) express the same state of affairs, only in different language.

Here, *weather_type* is a variable, representing some type of weather.[13] The reasons matching this pattern arise by the application of a valid rule. The rule can be represented as follows

rule(weather_report(*weather_type*),
    weather_type)

The rule has a condition weather_report(*weather_type*) and a conclusion *weather_type*. The use of lower-case characters shows that the rule is represented as a term: we treat a rule as an object that represents a relation between condition and conclusion. The fact that this rule is valid is expressed by the following sentence:

Valid(rule(weather_report(*weather_type*),
    weather_type)) (1)

The rule gives rise to a reason if it applies. In our example, the rule applies (initially) since Weather_report(rainy_day) is true. The fact that the rule above applies is expressed as

Applies(rule(weather_report(*weather_type*),
        weather_type),
    weather_report(rainy_day),
    rainy_day).

This sentence expresses that the rule with condition weather_report(*weather_type*) and conclusion *weather_type* applies on the basis of the fact weather_report(rainy_day).

3.2 Exclusionary reasons

A rule normally applies if its condition is satisfied. However, as we have seen, this is not always the case. For instance, a rule does not apply if the rule is excluded because of an exclusionary reason. We saw that 'The weather report on national television is not good at predicting the local weather' was an exclusionary reason against the applicability of the rule (1) above. This fact is expressed by the following sentence:

Reason(bad_local_prediction,
    excluded(rule(weather_report(*weather_type*),

---

[13] This suggests that a formal language with typed variables could be useful (see for instance Davis, 1993, p. 40ff. on many-sorted logic). We will not do this, in order to make the formalism not unnecessarily complicated.

        weather_type),
    weather_report(rainy_day),
    rainy_day))

As a result, the rule (1) is excluded:

Excluded(rule(weather_report(*weather_type*),
        weather_type),
    weather_report(rainy_day),
    rainy_day)

Since the rule is excluded, the rule is not applicable. In Reason-Based Logic, this is expressed as follows:

¬Applicable(rule(weather_report(*weather_type*),
        weather_type),
    weather_report(rainy_day),
    rainy_day)

In the example, the rule does not apply, and the sentence

Reason(weather_report(rainy_day),
    rainy_day)

is false. So far, there is no reason to conclude that it will rain.

3.3 Weighing reasons

Later in our story John and Mary had two reasons concerning the weather that Sunday:

Reason(belief_father(rainy_day),
    rainy_day)
Reason(cloudless_sky,
    ¬rainy_day)

So, Belief_father(rainy_day) is a reason for Rainy_day, while Cloudless_sky is a reason for ¬Rainy_day, i.e., a reason against Rainy_day. In such a case of conflicting reasons, the reasons must be weighed. John and Mary decide that Cloudless_sky as a reason against Rainy_day outweighs the reason Belief_father(rainy_day). This is expressed as:

```
Outweighs({cloudless_sky},
    {belief_father(rainy_day)},
    ¬rainy_day)
```

More precisely, this sentence expresses that the set of reasons containing only the reason Cloudless_sky for ¬rainy_day (i.e, against rainy_day) outweighs the set of reasons containing only the reason Belief_father(rainy_day) for rainy_day. Sets of reasons are needed since there can be several reasons pointing in the same direction.

3.4 Reasons concerning the application of a rule

We saw that there can be reasons for and against the application of a rule. In our example, John and Mary knew that if they would apply the rule (1) and as a result conclude that it will rain, their trip would no longer be enjoyable. That gives a reason against the application of the rule:

Reason(trip_no_longer_enjoyable,
    ¬applies(rule(weather_report(*weather_type*),
            weather_type),
        weather_report(rainy_day),
        rainy_day))

However, the condition of the rule is satisfied, since Weather_report(rainy_day) is true (after John and Mary hear the radio). The rule is this time not excluded, so it is applicable. If a rule is applicable, the fact that makes it applicable is a reason to apply the rule. So, there is also a reason for applying the rule:

Reason(weather_report(rainy_day),
    applies(rule(weather_report(*weather_type*),
            weather_type),
        weather_report(rainy_day),
        rainy_day))

John and Mary consider the reason not to apply the rule stronger:

Outweighs({trip_no_longer_enjoyable},
    {weather_report(rainy_day)},
    ¬applies(rule(weather_report(*weather_type*),
            weather_type),
        weather_report(rainy_day),
        rainy_day))

and they do not apply the rule.

4 Types of facts

In this section, we start with the formal definition of Reason-Based Logic.[14] We specify the types of facts concerning rules and reasons by defining a formal language in which the different types of facts can be expressed.

The language of Reason-Based Logic (RBL) is based on that of First-Order Predicate Logic (FOPL).[15] However, there are differences since the language of Reason-Based Logic must be appropriate to represent the types of facts concerning rules and reasons that we have encountered.

The main differences are that the language of Reason-Based Logic contains a number of special function and predicate symbols, and that there is a translation from sentences to terms.

As a result, terms and sentences must adhere to certain constraints. Therefore, after the definition of alphabets (section 4.1), we must distinguish between pre-terms and pre-sentences, not adhering to the constraints, and terms and sentences, adhering to the constraints. In section 4.2, pre-terms and pre-sentences are defined, analogous to terms and sentences of First-Order Predicate Logic. In section 4.3, we define the translation from sentences to terms. In section 4.4, we then define terms and sentences as pre-terms and pre-sentences adhering to certain constraints. Section 4.5 contains an overview of the types of facts.

4.1 Alphabets of Reason-Based Logic

The following definition shows that an alphabet of Reason-Based Logic is identical to an alphabet of First-Order Predicate Logic that contains some special-purpose function and predicate symbols.

Definition 1.

Function symbols are finite strings of symbols a, b, c, ..., z, A, B, C, ..., Z, _ starting with a lower-case.

Predicate symbols are finite strings of symbols a, b, c, ..., z, A, B, C, ..., Z, _ starting with an upper-case.

Variable symbols are finite strings of symbols *a*, *b*, *c*, ..., *z*, *A*, *B*, *C*, ..., *Z*, _ starting with a lower-case.

An *alphabet* of Reason-Based Logic is any set consisting of

---

[14] Several versions of Reason-Based Logic have been presented over the years, e.g., by Hage (1991, 1993, 1995), Hage and Verheij (1994a, b) and Verheij (1993, 1994, 1995e). The differences are mainly due to new insights or differences of focus. For instance, Hage (1995) has extended Reason-Based Logic to incorporate reasoning with goals, while Verheij (1994) used a limited version of Reason-Based Logic to get the formalism right. See also note 1.

[15] In the following we do not go into details of First-Order Predicate Logic, and assume that the reader has some familiarity with it. For instance, Van Dalen (1983) gives a good introduction to the syntax and semantics of First-Order Predicate Logic.

1. the function symbol rule with arity 2, plus any number of additional function symbols, each assigned a natural number denoting its arity,
2. the predicate symbols Reason with arity 2, Valid with arity 1, Excluded with arity 3, Applicable with arity 3, Applies with arity 3, and Outweighs with arity 3, plus any number of additional predicate symbols, each assigned a natural number denoting its arity,
3. variable symbols, and
4. the symbols (,), {, }, ¬, ∧, ∨, →, ∃, ∀, =, : and ,.[16]

Function and predicate symbols do not need to have a unique arity.

The smallest alphabet consists of the function symbol rule with arity 2, predicate symbols Reason with arity 2, Valid with arity 1, Excluded with arity 3, Applicable with arity 3, Applies with arity 3, and Outweighs with arity 3, no variable symbols, and the symbols (,), {, }, ¬, ∧, ∨, →, ∃, ∀, =, : and ,. The largest alphabet consists of all function predicate symbols (with all arities), all variable symbols, and the symbols (,), {, }, ¬, ∧, ∨, →, ∃, ∀, =, : and ,.

In the following, the definitions refer to a fixed alphabet of Reason-Based Logic.

4.2 Pre-terms and pre-sentences

Before we can define the terms and sentences of Reason-Based Logic, we need to define pre-terms and pre-sentences. These are defined in a similar way as the terms and sentences of First-Order Predicate Logic. (The terms and sentences of Reason-Based Logic have to adhere to certain additional constraints.) In the following definition, n denotes a natural number, n > 0, except when otherwise indicated.

Definition 2.

The set of *pre-terms* of Reason-Based Logic is the smallest set such that the following holds:

1. Any function symbol with arity 0 and any variable symbol is a pre-term.
2. If $term_1$, $term_2$, ..., and $term_n$ are pre-terms and *function* is a function symbol with arity n, then $function(term_1, term_2, \ldots term_n)$ is a pre-term.
3. If $term_1$, $term_2$, ..., and $term_n$, with $n \geq 0$, are pre-terms, then $\neg term_1$, $(term_1 \wedge term_2)$, $(term_1 \vee term_2)$ and $\{term_1, term_2, \ldots, term_n\}$ are pre-terms.

For convenience, we use the same typographical style for variable symbols and metavariables. The role of the pre-terms of the forms $\neg term_1$, $(term_1 \wedge term_2)$, $(term_1 \vee term_2)$ and $\{term_1, term_2, \ldots, term_n\}$ will be explained below (section 4.3).

Three examples of pre-terms are:

[16] Here the comma ‘,’ (of the normal text font) is used to separate the symbols of the alphabet, and the comma ‘,’ (of the formula font) is one of the symbols.

mary
father(john)
rule(weather_report(*weather_type*),
weather_type)

Definition 3.

The set of *pre-formulas* is the smallest set such that the following hold:

1. If $term_1$ and $term_2$ are pre-terms, then $term_1 = term_2$ is a pre-formula.
2. Any predicate symbol with arity 0 is a pre-formula.
3. If $term_1$, $term_2$, ..., and $term_n$ are pre-terms and *Predicate* is a predicate symbol with arity n, then *Predicate*($term_1$, $term_2$, ... $term_n$) is a pre-formula.
4. If $Formula_1$ and $Formula_2$ are pre-formulas, then $\neg Formula_1$, $(Formula_1 \wedge Formula_2)$, $(Formula_1 \vee Formula_2)$ and $(Formula_1 \rightarrow Formula_2)$ are pre-formulas.
5. If *Formula* is a pre-formula and *x* is a variable symbol, then $\exists x$: *Formula* and $\forall x$: *Formula* are pre-formulas.

A *pre-atom* is a pre-formula of one the forms *Predicate*, $term_1 = term_2$, or *Predicate*($term_1$, $term_2$, ... $term_n$). A *pre-literal* is a pre-atom or a pre-atom preceded by $\neg$. A *pre-sentence* is a pre-formula without free variables.[17]

We use the ordinary conventions to reduce the number of brackets in formulas.

Three examples of pre-sentences and pre-formulas are:

Is_thief(mary)
Predicts(local_weather_report, rainy_day)
Valid(rule(weather_report(*weather_type*),
weather_type))

Shortly, we will see that not all pre-terms and pre-sentences are terms and sentences of Reason-Based Logic. We return to this issue in section 4.4.

4.3 A translation from sentences to terms

As mentioned in section 3.1, in Reason-Based Logic, we do not only need to express states of affairs as sentences, but also to refer to them in other sentences. In the formal language, we use a translation from (pre-)sentences to (pre-)terms in order to refer to sentences.[18]

We use a simple translation: to obtain the pre-term that corresponds to a (quantifier free) pre-sentence, the first upper-case character of each predicate

[17] Free variables are defined as usual.

[18] This is an often-encountered technique, known as *reification*. For other examples, we refer to the overview of meta-languages, reflection principles and self-reference by Perlis and Subrahmanian (1994).

symbol in the pre-sentence is replaced by the same character in lower-case. By the choice of alphabet and the definition of terms, the result of this translation is always a pre-term.

For example, the pre-sentence

Is_thief(mary)

translates to the term

is_thief(mary).

As the definition of pre-terms shows, the logical connectives are treated as if they also are function symbols. In this way, the translation can be kept as simple as it is now. For example, the pre-sentence

Is_guilty(mary) $\wedge$ $\neg$Punish(mary)

translates to the term

is_guilty(mary) $\wedge$ $\neg$punish(mary).

To stay as close as possible to the usual notation of sentences, the logical connectives are *infix* function symbols. For instance, instead of writing terms of the form $\wedge(term_1, term_2)$, we write $term_1 \wedge term_2$.

Of course not all terms should be translations of sentences. For instance, the terms mary and father(john) do not correspond to sentences Mary and Father(john). Therefore, we should divide the set of terms into two types, namely those that correspond to sentences, and those that do not. As a result, only a subset of all strings of characters beginning with an upper-case can be predicate symbols. For convenience, we will not explicitly define such a subset, but assume that any string of characters beginning with an upper-case that we encounter is in this subset.

The translation easily extends to metavariables for (pre-)sentences and (pre-)terms, as follows. Metavariables for pre-sentences will be denoted as strings of italic characters beginning with an upper-case character, e.g., *Fact*. Metavariables for pre-terms will be denoted as strings of italic characters beginning with a lower-case (just as the variables of the logical language), e.g., *fact*. Matching metavariables for pre-sentences and pre-terms, such as *Fact* and *fact*, represent a sentence and its translation to a term. This extended translation will turn out to be crucial in several of the coming definitions.

4.4 Terms and sentences

In Reason-Based Logic, there are function and predicate symbols that play a special role. There are constraints on their use. Formally, we define terms and formulas as pre-terms and pre-formulas that adhere to a number of constraints.

Definition 4.

A *term* of Reason-Based Logic is a pre-term that adheres to the following constraints:

1. If rule(*condition*, *conclusion*) is a pre-term, *Condition* must be a disjunction of conjunctions of pre-literals and *Conclusion* a pre-literal.
2. If {$fact_1$, $fact_2$, ..., $fact_n$} is a pre-term, $Fact_1$, $Fact_2$, ..., and $Fact_n$ must be disjunctions of conjunctions of pre-literals.

In this definition, we use the translation from sentences to terms for the first time in a formal definition. For instance, *Condition* denotes a sentence that translates to the term denoted by *condition*.

Definition 5.

A *formula* of Reason-Based Logic is a pre-formula *Formula* that adheres to the following constraints:

1. All pre-terms that occur in *Formula* must be terms.
2. If *Formula* has the form

   Reason(*fact*, *state_of_affairs*),
   Valid(rule(*condition*, *conclusion*)),
   Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*),
   Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*),
   Applies(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*), or
   Outweighs($reasons_1$, $reasons_2$, *state_of_affairs*),

   then the following must hold:

   a. *Fact*, *State_of_affairs*, $Reason_1$, $Reason_2$, ... and $Reason_n$ must be pre-sentences, i.e., do not contain free variables.
   b. *Fact* must be a disjunction of conjunctions of pre-literals and must be an instance of *Condition* under some substitution σ, and *State_of_affairs* must be a pre-literal that is an instance of *Conclusion* under the same substitution σ.
   c. The (pre-)terms $reasons_1$ and $reasons_2$ must both have the form {$fact_1$, $fact_2$, ..., $fact_n$}, with $n \geq 0$.

Atoms and *literals* are formulas that are pre-atoms and pre-literals, respectively. *Sentences* are pre-formulas that only contain free variables in occurrences of terms of the form rule(*condition*, *conclusion*).

Definition 6.

A *language* of Reason-Based Logic is the set of formulas belonging to some alphabet of Reason-Based Logic.

4.5 Overview of the types of facts

As we saw in section 3, in Reason-Based Logic, a number of function and predicate symbols are used to express types of facts concerning rules and reasons. Below we provide an overview of these function and predicate symbols and their use.

- rule(*condition*, *conclusion*)

  Since we treat rules as objects, rules are represented as terms in Reason-Based Logic. In this way it is possible to express facts about rules. A term denoting a rule has the form rule(*condition*, *conclusion*). Here *condition* and *conclusion* are terms with free variables. The formula *Condition* that translates to the term *condition* must be a disjunction of conjunctions of one or more literals. In other words, *Condition* is quantifier free and in disjunctive normal form. An instance of *Condition* is a possible reason for a matching instance of *Conclusion*. The formula *Conclusion* that translates to the term *conclusion* must be a literal.

- {$fact_1$, $fact_2$, ..., $fact_n$} (for n = 1, 2, ...)

  These symbols are used to refer to the sets of facts that are reasons for some conclusion. We use an unusual syntax of terms to stay as close as possible to the normal notation of sets. The term {thief(mary), minor(mary)} refers to the set of the two reasons expressed by the sentences Thief(mary) and Minor(mary). The term { } (without arguments) is used to denote an empty set of reasons.

  There is a problem here with different terms that denote identical sets, such as {thief(mary), minor(mary)} and {minor(mary), thief(mary)}. Axioms should be included in Reason-Based Logic such that formulas that only differ in such equivalent terms for sets are equivalent. We will not do this explicitly.

  We do not consider infinite sets of reasons.

- Reason(*fact*, *state_of_affairs*)

  A sentence of this form expresses that the fact referred to by the term *fact* is a reason for the state of affairs referred to by the term *state_of_affairs*. The sentence *Fact* (that translates to the term *fact*) must be a disjunction of conjunctions of literals, and *State_of_affairs* (that translates to the term *fact*) a literal. If *State_of_affairs* is an atom *Atom*, *Fact* is a reason for *Atom* and a reason against ¬*Atom*; similarly, if *State_of_affairs* is a negated atom ¬*Atom*, *Fact* is a reason for ¬*Atom* and a reason against *Atom*.

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

  A sentence of this form expresses that the rule with condition *condition* and conclusion *conclusion* is valid.

- Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)

  A sentence of this form expresses that the rule with condition *condition* and conclusion *conclusion* is excluded, for the instance *Fact* of the rule's condition *Condition*. *Fact* must be an instance of *Condition*, and *State_of_affairs* an instance of *Conclusion*.

- Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)

  A sentence of this form expresses that the rule with condition *condition* and conclusion *conclusion* is made applicable by the fact expressed by the term *fact*. If a rule is applicable, it may give rise to a reason for the state of affairs expressed by the term *state_of_affairs*. *Fact* must be an instance of one of the disjuncts of *Condition*, and *State_of_affairs* an instance of *Conclusion*.

- Applies(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)

  A sentence of this form expresses that the rule with condition *condition* and conclusion *conclusion* applies on the basis of the fact expressed by *fact* and therefore generates a reason for the state of affairs expressed by *state_of_affairs*. *Fact* must be an instance of *Condition*, and *State_of_affairs* an instance of *Conclusion*. The predicate *Applies* should not be confused with the predicate *Applicable*. The difference in meaning (introduced in the sections 1 and 3) is made precise in the next section.

- Outweighs(*reason_pro*, *reasons_con*, *state_of_affairs*)

  A sentence of this form expresses that the reasons in the set referred to by the term *reasons_pro* outweigh the reasons in the set referred to by the term *reasons_con* (as reasons concerning *state_of_affairs*). The terms *reasons_pro* and *reasons_con* must both have the form {$fact_1$, $fact_2$, ..., $fact_n$}, where $n \geq 0$. Each sentence $Fact_i$ must be a disjunction of conjunctions of literals (for each i from 1 to n), and *State_of_affairs* a literal. The reasons in *reasons_pro* are reasons for *State_of_affairs*, and the reasons in *reasons_con* are reasons against *State_of_affairs*. Equivalently, if *Not_state_of_affairs* is the literal that is the opposite of *State_of_affairs*, the reasons in *reasons_pro* are reasons against *Not_state_of_affairs*, and the reasons in *reasons_con* are reasons for *Not_state_of_affairs*.

5 Relations between facts

In this section, we describe the relations that hold between the described facts concerning rules and reasons. We do it in terms of the truth values of the corresponding sentences. The basis is again First-Order Predicate Logic.[19] The relations that hold between facts (in terms of the truth values of sentences that express the facts) as defined by First-Order Predicate Logic also hold in Reason-Based Logic. For instance, the following relations hold:

NOT
For all sentences *State_of_affairs*,
Either *State_of_affairs* is true or ¬*State_of_affairs* is true.

AND
For all sentences $State\_of\_affairs_1$ and $State\_of\_affairs_2$,
$State\_of\_affairs_1$ is true and $State\_of\_affairs_2$ is true if and only if
$State\_of\_affairs_1 \wedge State\_of\_affairs_2$ is true.

OR
For all sentences $State\_of\_affairs_1$ and $State\_of\_affairs_2$,
$State\_of\_affairs_1$ is true or $State\_of\_affairs_2$ is true if and only if
$State\_of\_affairs_1 \vee State\_of\_affairs_2$ is true.

The relations that hold between sentences that are typical for Reason-Based Logic are defined in a similar way. They are called VALIDITY, EXCLUSION, APPLICABILITY, APPLICATION, WEIGHING, and WEIGHING_AXIOMS.[20] We assume in the following that all mentioned sentences are well-formed, i.e., are sentences of the language of Reason-Based Logic.

VALIDITY
For all sentences *Condition*, *Conclusion*, *Fact* and *State_of_affairs*,
If Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*),
Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) or
Applies(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) is true, then
Valid(rule(*condition*, *conclusion*)) is true.

---

19 For convenience, we will not as usual define the relations between facts in terms of structures and models, but in terms of truth values of sentences. Such a definition can be given, but does not provide additional insight, while the formalism becomes more complex.

20 These relations could also be given as a set of axioms. We have chosen the present form in order to stress that in Reason-Based Logic the standard logical connectives, such as ¬ and ∧, are not treated differently from the non-standard logical constants, such as Valid and Applicable.

Informally, VALIDITY says that a rule can only be excluded, be applicable, or apply if it is valid.

EXCLUSION

For all sentences *Fact* and *State_of_affairs*,

If *Fact* and Valid(rule(*condition*, *conclusion*)) are true, then either Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) or Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) is true.

Informally, EXCLUSION says that a rule is either excluded or applicable if its condition is satisfied. Here *Fact* stands for the fact that satisfies the condition of the rule.

APPLICABILITY

For all sentences *Fact* and *State_of_affairs*,

a. Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) is true if and only if Reason(*fact*, applies(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*)) is true.
b. If Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) is true, then *Fact* is true.

Informally the first part of APPLICABILITY says that if and only if a rule is applicable, the fact that makes the rule applicable is a reason to apply the rule. The second part says that a rule can only be applicable if its condition is satisfied. Again, *Fact* stands for the fact that satisfies the condition of the rule.

APPLICATION

For all sentences *Fact* and *State_of_affairs*,

There are terms *condition* and *conclusion*, such that Applies(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) is true if and only if Reason(*fact*, *state_of_affairs*) is true.

Informally this relation says that if and only if a rule applies, the fact that makes the rule applicable is a reason for the rule's (instantiated) conclusion. or, equivalently, a reason against the opposite of the rule's conclusion.

Notice the difference between a rule's being applicable and its being applied. If a rule is applicable, this only indicates that there is a reason for applying the rule (see APPLICABILITY, part a). In general, there can also be reasons against applying a rule.

WEIGHING

For all sentences Pro_1, Pro_2, ..., Pro_n (for some natural number n), Con_1, Con_2, ..., Con_m (for some natural number m), *State_of_affairs*, and its opposite *Not_state_of_affairs*,

> If Reason(*pro*$_1$, *state_of_affairs*), Reason(*pro*$_2$, *state_of_affairs*), ..., Reason(*pro*$_n$, *state_of_affairs*), Reason(*con*$_1$, *not_state_of_affairs*), Reason(*con*$_2$, *not_state_of_affairs*), ..., Reason(*con*$_m$, *not_state_of_affairs*), and also Outweighs({*pro*$_1$, *pro*$_2$, ..., *pro*$_n$}, {*con*$_1$, *con*$_2$, ..., *con*$_m$}, *state_of_affairs*) is true, then *State_of_affairs* is true, or there is a term *con*, different from *con*$_1$, *con*$_2$, ..., and *con*$_m$, such that Reason(*con*, *not_state_of_affairs*) is true.

Informally the first part of this relation says that reasons make a conclusion true if the pros outweigh the cons, provided that no con is overlooked. It is allowed that one or more of the pros is overlooked: if a subset of the pros already suffices to outweigh all cons, the conclusion certainly follows if there are even more pros.[21] It may seem that a similar relation between facts is required for the case that the cons outweigh the pros. However, since in Reason-Based Logic a reason against a state of affairs is just a reason for the opposite state of affairs, the relation above suffices.[22]

WEIGHING_AXIOMS

For all sentences *Fact*$_1$, *Fact*$_2$, ..., *Fact*$_n$ (for some positive natural number n), *State_of_affairs*, and its opposite *Not_state_of_affairs*, and all terms *pros* and *cons*,

a. Outweighs(*pros*, *cons*, *state_of_affairs*) and Outweighs(*cons*, *pros*, *not_state_of_affairs*) are not both true.
b. If Reason(*fact*$_1$, *state_of_affairs*), Reason(*fact*$_2$, *state_of_affairs*), ..., Reason(*fact*$_n$, *state_of_affairs*) are true, then Outweighs({*fact*$_1$, *fact*$_2$, ..., *fact*$_n$}, { }, *state_of_affairs*) is true.

The first part of this relation says that the pros as reasons for *state_of_affairs* cannot outweigh the cons and the other way around at the same time. However, the first weighing axiom does not make it impossible that ¬Outweighs(*pros*, *cons*, *state_of_affairs*) and ¬Outweighs(*cons*, *pros*, *state_of_affairs*) are both true.

Reason-Based Logic does in general not determine which set of reasons outweighs another set. However, for the case that all reasons point in the same direction, i.e., all reasons are either pros or cons, the second part of the relation gives the result: any non-empty set of reasons outweighs the empty one.

---

21 This is due to the *accrual of reasons*, a term used by Pollock (1991, p. 51). Accrual is discussed more extensively later on.

22 In other versions of Reason-Based Logic (e.g., Hage and Verheij, 1994a), the two cases that the pros outweigh the cons and that the cons outweigh the pros, are formally distinguished, even though there is no conceptual distinction. In the version of Reason-Based Logic described by Verheij (1995e), this is acknowledged, and the two cases are no longer formally distinguished.

6 Conclusions following from given premises

Although it is not strictly part of the semantics of rules and reasons, we discuss in this section which conclusions follow from given premises. The given set of premises, representing a theory of rules and reasons, is called a theory of Reason-Based Logic.

The simplest approach is to define which conclusions deductively follow from a given theory analogous to First-Order Predicate Logic, as follows:

Definition 7. (RBL-deduction)

A *theory* of Reason-Based Logic is any set of sentences (in a given language of Reason-Based Logic). A conclusion *Conclusion deductively follows* from a theory T, if the truth of the sentences in T follows from the truth of the sentence *Conclusion*, using the relations between facts of Reason-Based Logic.[23]

Definition 7 extends deduction in First-Order Predicate Logic, and allows that conclusions are drawn on the basis of the relations between facts that hold in Reason-Based Logic. It is possible to define a set of deduction rules, in the style of First-Order Predicate Logic's natural deduction, that are sound and complete with respect to this deductive consequence relation. However, this consequence relation turns out to be weak, and intuitively attractive types of reasoning on the basis of reasons are not captured by RBL-deduction.

As a result, we do not devote much attention to the deductive consequence relation, and focus on a more interesting *nonmonotonic* consequence relation.

We give an example of a type of reasoning that is not captured by RBL-deduction: the conclusion Rainy_day does not follow from the theory that consists of the two sentences

Weather_report(rainy_day)
Valid(rule(weather_report(*weather_type*),
weather_type))

Intuitively, simply applying the rule, the condition of which is satisfied, leads to the conclusion Rainy_day. The difficulty is hidden in the world 'simply': the rule does not simply apply, since for the rule to apply several semantical constraints must be met. As a result, not in all circumstances in which the theory above is true, the rule actually applies. Formally, the sentence

Applies(rule(weather_report(*weather_type*),
weather_type),

[23] Normally, which conclusions follow from a theory is defined in terms of the *models* of the theory. But see note 19.

```
    weather_report(rainy_day),
    rainy_day)
```

is not always true. For instance, it can be the case that the rule is excluded, i.e., in which

```
Excluded(rule(weather_report(weather_type),
        weather_type),
    weather_report(rainy_day),
    rainy_day)
```

is true. Then the rule is not applicable and normally not applied.

Intuitively, however, it seems most natural that the rule is not excluded, since there is no information in the theory that makes it excluded. Therefore, it seems natural to allow the following type of reasoning:

> If the condition of a rule is satisfied, then it follows that the rule is applicable, unless it follows that the rule is excluded.

This type of reasoning is an example of a *nonmonotonic* rule of inference. It is called nonmonotonic, since it can be the case that conclusions based on it must be retracted because of newly inferred facts. For instance, it may seem now that a rule is not excluded with respect to the currently inferred facts, but later it may be inferred that the rule is excluded after all. This is in contrast with the usual monotonic rules of inference. Once a conclusion based on monotonic rules of inference is established, it never has to be retracted on the basis of newly inferred facts.

The problem with nonmonotonic rules of inference is that they can only be safely used to draw conclusions if one knows all consequences of a theory in advance. This is in conflict with the step by step construction of the set of consequences of a theory: starting from the premises in the theory conclusions are added step by step by drawing new conclusions using the rules of inference. As a result, the complete set of conclusion following from a theory is only known after all steps have been completed.

Many approaches to deal with nonmonotonic rules of inference have been proposed. Ginsberg (1987), Lukaszewicz (1990) and Gabbay *et al.* (1994b) have given overviews of such research. We present an approach based on extensions that is related to ideas that go back to Reiter's Default Logic (1980, 1987).

In the definition of the nonmonotonic consequences of a theory we use a set of sentences that can be regarded as a guess in advance of the set of consequences. The nonmonotonic rule of inference mentioned above is then read in a slightly different way, by referring to this guess:

> If the condition of a rule is satisfied, then it follows that the rule is applicable, unless it is guessed that it follows that the rule is excluded.

Let now T be a theory, and S a set of sentences, that represents our guess of consequences following from the theory T. We will define which conclusions follow from the theory T relative to the guess set S. The following rule of inference (related to the relation between facts EXCLUSION of section 5) holds in Reason-Based Logic:

EXCLUSION*

For all sentences *Fact* and *State_of_affairs*,

If *Fact* and Valid(rule(*condition*, *conclusion*)) follow from T relative to S, then Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) follows from T relative to S, unless Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*) is an element of S.

This rule of inference says that if it follows that the condition of a rule is satisfied, it follows that the rule is applicable, unless it is guessed that the rule is excluded.

There is a second type of reasoning that is intuitively attractive, but is not captured by RBL-deduction. Informally:

> If it follows that all derivable pros outweigh all derivable cons, the conclusion of the pros follows. If it follows that all derivable cons outweigh all derivable pros, the conclusion of the cons follows.

This type of reasoning is however also an example of a nonmonotonic rule of inference. Since it refers to all derivable pros and cons, one has to know the whole set of conclusions in advance. Again we use the fixed guess set S to avoid the difficulties. Instead of using all derivable pros and cons, the following rule of inference (related to the relation between facts WEIGHING of section 5) uses all reasons in the guess set S.

WEIGHING*

For all sentences Pro_1, Pro_2, ..., Pro_n (for some natural number n), Con_1, Con_2, ..., Con_m (for some natural number m), *State_of_affairs*, and its opposite *Not_state_of_affairs*,

If Reason(pro_1, *state_of_affairs*), Reason(pro_2, *state_of_affairs*), ..., Reason(pro_n, *state_of_affairs*), Reason(con_1, *not_state_of_affairs*), Reason(con_2, *not_state_of_affairs*), ..., Reason(con_m, *not_state_of_affairs*), and also Outweighs($\{pro_1, pro_2, ..., pro_n\}$, $\{con_1, con_2, ..., con_m\}$, *state_of_affairs*) follow from T relative to S, then *State_of_affairs* follows from T relative to S, unless there is a term *con*, different from con_1, con_2, ..., and con_m, such that Reason(*con*, *not_state_of_affairs*) is an element of S.

Finally, the conclusions that deductively follow from a theory also follow from T relative to S:

RBL-DEDUCTION
1. All elements of T follow from T relative to S.
2. All sentences that (deductively) follow from sentences that follow from T relative to S follow from T relative to S.

The conclusions that follow from a theory T relative to a guess set S can now be defined as in First-Order Predicate Logic by a recursive definition using the rules of inference EXCLUSION*, WEIGHING* and RBL-DEDUCTION. The problems of such a recursive definition for nonmonotonic rules of inference have been avoided by translating these rules to monotonic rules of inference relative to the fixed set S. We have the following ordinary recursive definition of the conclusions that follow from a theory relative to a guess set:

Definition 8. (S-consequences)

A *guess set* of Reason-Based Logic is any set of sentences (in a given language of Reason-Based Logic). For any theory T and any guess set S, the set of conclusions that *follow from* T *relative to* S is the smallest set of sentences, such that EXCLUSION*, WEIGHING* and RBL-DEDUCTION hold. The conclusions that follow from T relative to the guess set S, are the S-*consequences* of T.

If T is a theory and S is a guess set, there are two cases in which the guess set S is not acceptable as a set of nonmonotonic consequences of T. First the guess set can be too small: there are S-consequences of T that are not in the guess set S. Second the guess set can be too large: not all sentences in the guess set S are S-consequences of T. So, a guess set is a set of nonmonotonic consequences of T if and only if the guess set is equal to the set of consequences relative to the guess set. A set of nonmonotonic consequences is usually called an extension. We get the following fixed-point definition:

Definition 9. (extensions)

For any theory T, a set of sentences E is an *extension* if and only if E is equal to the set of E-consequences of T.[24]

[24] One can see that this definition of extension corresponds to Reiter's (1980, 1987) if one reads the rules EXCLUSION* and WEIGHING* as defaults. For instance, EXCLUSION* corresponds to defaults with prerequisite *Fact* $\wedge$ Valid(rule(*condition*, *conclusion*)), justification Excluded(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*), and consequent Applicable(rule(*condition*, *conclusion*), *fact*, *state_of_affairs*). Our set of S-consequences of a theory T corresponds to Reiter's set $\Gamma_T(S)$. Of course, several unessential technical adaptations are necessary, such as using an RBL language.

A theory does not necessarily have an extension, and, if it has one, the extension is not necessarily unique. For instance, the theory that consists of the sentence

Valid(true, excluded(*condition*, *conclusion*))

has no extension. The theory that consists of the four sentences

A
B
Valid(rule(a, excluded(rule(b, *conclusion*))))
Valid(rule(b, excluded(rule(a, *conclusion*))))

has two extensions, namely one in which the first rule is excluded and the second rule applies, the other in which the first rule applies and the second rule is excluded.

Theories that have no or several extensions contain a paradox resembling the well-known paradoxes of self-reference. In Reason-Based Logic, such paradoxes are possible because of the translation from sentences to terms (as defined in section 4.3). We consider it the task of theories in Reason-Based Logic, rather than of the consequence relation of Reason-Based Logic, to avoid these paradoxes.