Chapter 3

Reason-Based Logic and law

This chapter contains examples of Reason-Based Logic, taken from the field of law. The examples illustrate the basic elements of Reason-Based Logic, and give applications to the theory of legal reasoning.

We start with a discussion of the apparent dichotomy of reasoning with rules and reasoning with principles (section 1). Our claim is that the seeming difference is merely a matter of degree. We support this claim by giving an integrated view on rules and principles (section 2). Before the formal elaboration of this view in Reason-Based Logic (section 7), we discuss how isolated rules/principles, the weighing of reasons, exceptions and conflicts can be modeled in Reason-Based Logic (sections 3, 4, 5 and 6, respectively). We end the chapter with an application of our view on rules and principles to reasoning by analogy (section 8). We show how this view gives rise to three different ways of reconstructing reasoning by analogy.[1]

1 Reasoning with rules vs. reasoning with principles

There seem to be two types of reasoning:

- *Reasoning with rules*
  A rule is applied if its condition is satisfied. If a rule is applied, its conclusion follows directly.

- *Reasoning with principles*
  In contrast with a rule, a principle only gives rise to a reason for its conclusion if it applies. Moreover, there can be other applying principles that give rise to both reasons for and reasons against the conclusion. As a result, a conclusion only follows by weighing the pros and cons.

[1] The sections 1, 3, 7 and 8 of this chapter are based on the papers by Verheij and Hage (1994) and Verheij (1996b).

For instance, Dworkin (1978, p. 22ff. and p. 71ff.) has made a strict distinction between rules and principles in the field of law. According to Dworkin, rules have an all-or-nothing character, while principles have a dimension of weight or importance. An example of a typical rule, he says, is the proposition 'A will is invalid unless signed by three witnesses'. An example of a typical principle is 'No man may profit from his own wrong'.[2]

There are at least three seeming differences between rules and principles. The first is that rules lead directly to their conclusion if they are applied, while principles lead to their conclusion in two steps: first principles give rise to reasons, then these reasons are weighed.

The second difference between rules and principles appears in the case of a conflict. In case of conflicting rules, that is rules with incompatible conclusions that apply to a single case, the rules lead directly to their conclusions, and therefore to a contradiction. In case of conflicting principles, i.e., if there are principles with incompatible conclusions that apply to a single case, no such problems occur. The application of conflicting principles only leads to reasons that plead for incompatible conclusions, so no contradiction is involved. In such cases, a conflict can involve several distinct reasons, some of which plead for a conclusion, others against it. Weighing the pros and cons determines the final conclusion.

The third difference is that rules lead to their conclusion in isolation, while principles interact with other principles. For instance, additional reasons arising from other principles can influence the result of the weighing of reasons.

These differences are summarized in Table 1.

	Rule	**Principle**
Application	Conclusion	Reason
Conflict	Contradiction	Weighing
Other rules/principles	Independent	Dependent

Table 1: The seeming differences between rules and reasons

This leads to the question whether rules and principles are logically different. There is no agreement. For instance, Dworkin has a strong opinion:

> 'The difference between legal principles and legal rules is a logical distinction' (Dworkin, 1978, p. 24)

[2] As Soeteman (1991, p. 33) notes, the usage of the terms 'rule' and 'principle' is not at all uniform. For instance, 'Ne bis in idem' is called a principle, but has a rule-like nature, while 'A contract must be executed in good faith' is a principle-like rule. Here, we do not deal with the usage of the terms 'rule' and 'principle', but with the nature of rules and principles.

Soeteman (1991), in his discussion of rules and principles, takes an apparently opposite stand:

> 'I know of no difference in logical structure between rules and principles.' (Soeteman, 1991, p. 34)[3]

Indeed, there are clear similarities between rules and principles. We mention two of them. First, rules and principles both are basically a connection of some sort between a *condition* and a *conclusion*. The difference is only that, in the case of a rule, the connection seems stronger than in the case of a principle.

Second, for a rule or principle *in isolation* the differences disappear. In isolation, the conclusion of both a rule and a principle follows if the condition is satisfied.

Because of these similarities, we claim that the seeming differences between rules and principles are merely a matter of degree. There is no clear border between reasoning with rules and principles. They are the two extremes of a spectrum.[4] We support our claim by giving an integrated representation of rules and principles in Reason-Based Logic in section 5.[5]

Some preliminaries are required. In the next section we informally discuss our integrated view on rules and principles. Then we discuss how isolated rules/principles, the weighing of reasons, exceptions and conflicts can be represented in Reason-Based Logic (sections 3, 4, 5 and 6, respectively).

2 An integrated view on rules and principles

Our integrated view on rules and principles is based on two main assumptions:

- Both rules and principles give rise to reasons if they are applied.
- The differences between reasoning with rules and principles result from different types of relationships with other rules and principles, which may interfere.

As a basic example of the role of the relationships between rules and principles, we discuss a rule and its underlying principles (section 2.1). Then we discuss our view on a typical rule (section 2.2), a typical principle (section 2.3), and a hybrid rule/principle (section 2.4).

---

3 Translated from the original in Dutch: 'Ik ken (...) geen verschil in logische structuur tussen regels en beginselen'.

4 Soeteman (1991) and Sartor (1994, p. 189) make similar claims. However, our integrated view is more explicit, and can explain the intuitive differences (see section 7).

5 By the formal elaboration, the view can be applied to the use of computers as tools in the field of law (cf. Van den Herik, 1991).

2.1 A rule and its underlying principles

A basic example of the relationships between rules and principles occurs when a rule has underlying principles.

For instance, if the legislator makes a legal rule, this is often based on a decision in which several factors are taken into account. These factors, or to use an already familiar term, reasons, are based on other rules and principles. If these reasons are in conflict, the legislator decides (either explicitly or implicitly) how they have to be weighed. We say that the rules and principles taken into account by the legislator *underlie* the newly made legal rule. In Figure 1, the situation is depicted. The principles underlying the rule that can lead to a reason for the conclusion of the rule are indicated as pro-principles, those that can lead to a reason against the conclusion are indicated as con-principles.

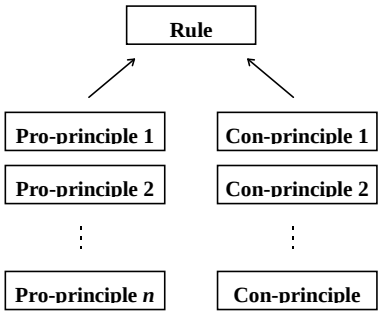

Figure 1: A rule and its underlying principles

As an example, we take the legal rule from Dutch civil law that sale of a house should not terminate an existing rent contract (Art. 7A:1612 BW).[6] This rule has, for instance, the following two underlying principles:

1. Somebody who lives in a house should be protected against measures that threaten the enjoyment of the house
2. Contracts only bind the contracting parties.

The first pleads against termination of an existing rent contract; the second pleads for termination since the new owner of the house does not have a contract with the person (or persons) living in the house. As a result, there is (at least) one underlying pro-principle, and one underlying con-principle.

Let us see what happens if the legal rule applies. Of course, its principles should normally not be applicable too since they have already been considered by the

6 This example is also discussed by Prakken (1993, pp. 22-23) and Verheij and Hage (1994), in the context of analogy. The discussion here is largely taken from the latter. We return to the example in section 8 when dealing with analogy.

legislator. We say that the legal rule when it applies *replaces* its underlying principles. As a result, if the rule of Art. 7A:1612 BW applies, its two underlying principles should not be applicable. The situation is shown in Figure 2.

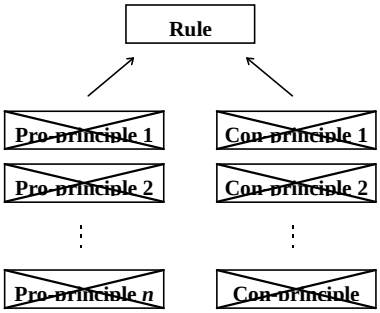

Figure 2: A rule replaces its underlying principles if it applies

If the rule did not replace its underlying principles, several reasons would arise that already had been taken account in the rule itself. However, because of the special relationships of the rule with its underlying principles, the principles should not be applicable.

2.2 A typical rule

In general, the relations between rules and principles are less clear than in the case of a rule and its underlying principles. These relationships can for instance be determined by the weight or importance of a rule or principle, or by the degree of pro- or con-ness. In Figure 3, we have suggested a set of interfering rules and principles.

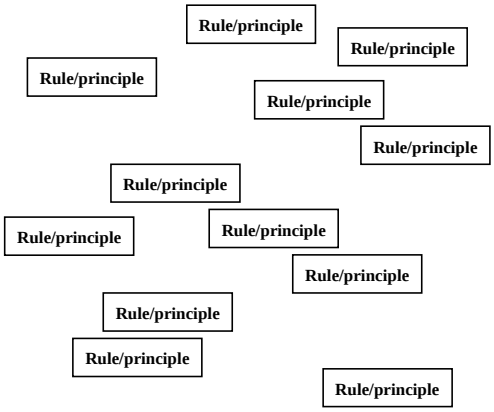

Figure 3: A set of interfering rules and principles

Assume that the rule/principle in the upper left corner is in fact a typical rule. In our view on rules and principles, if this typical rule applies, it blocks all interfering rules/principles. This situation is shown in Figure 4.

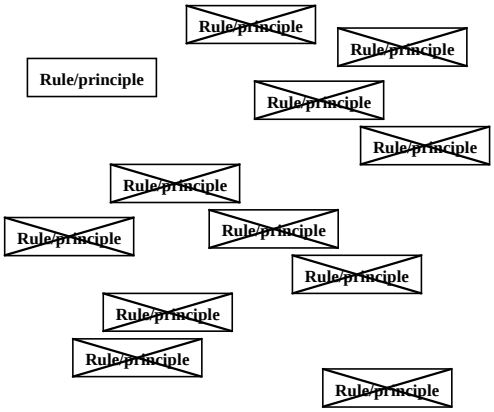

Figure 4: A typical rule applies

As a result, the conclusion of the rule follows directly.

2.3 A typical principle

If the rule/principle in the upper left corner were a typical principle, it would not block any of the interfering rules/principles in case it applies. The situation is shown in Figure 5.

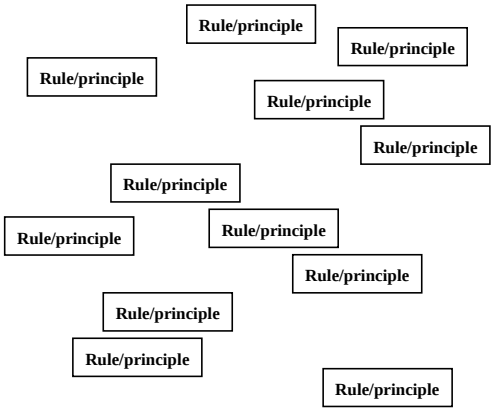

Figure 5: A typical principle applies

As a result, the conclusion of the principle does not follow directly, but only after weighing the reasons arising from the other rules/principles.

2.4 A hybrid rule/principle

Typical rules and typical principles are the extreme cases. Most rules/principles are hybrid: they are neither a typical rule, nor a typical principle. A hybrid rule/principle blocks some, but not all interfering rules/principles. The situation that the rule/principle in the upper left corner were a hybrid rule/principle and applies is shown in Figure 6.

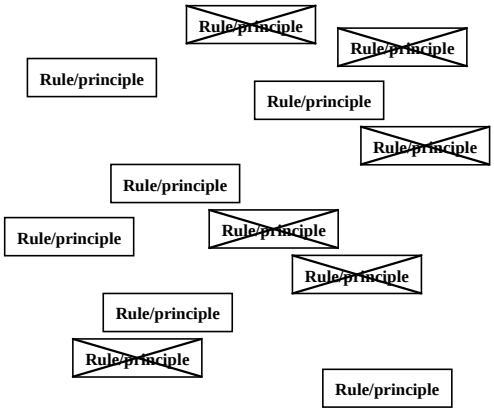

Figure 6: A hybrid rule/principle applies

As a result, the conclusion of the hybrid rule/principle does not follow directly, but only after weighing the reasons arising from the other rules/principles, that are not blocked.

In section 7, this informal sketch of an integrated view on rules and principles will be formalized in Reason-Based Logic. As preliminaries, we discuss how isolated rules/principles, the weighing of reasons, exceptions and conflicts are modeled in Reason-Based Logic (sections 3, 4, 5 and 6, respectively).

3 An isolated rule/principle in Reason-Based Logic

We start our discussion of rules and principles in Reason-Based Logic with the case of an isolated rule/principle. This will be spelled out in detail to illustrate the main elements of Reason-Based Logic.

As an example we use the legal rule that a person driving a car after drinking too much alcohol should be fined a considerable amount of money. (It does not matter that we use an isolated *rule* as an example, since in our view there is no

difference in representation between an isolated rule and an isolated principle.) Assume that we have:

> A person driving a car after drinking too much alcohol should be fined a considerable amount of money.
> John is driving his car after drinking too much alcohol.

If we interpret the first sentence as a rule, the application of this rule must lead to the conclusion:

> John should be fined a considerable amount of money.

This can be represented by the following three RBL sentences:[7]

```
Valid(rule(driving_with_alcohol(person),
    should_be_fined(person)))
Driving_with_alcohol(john)
Should_be_fined(john)
```

For this representation it does not matter whether the RBL rule concerning driving with alcohol stems from a rule or from a principle: both rules and principles are represented in Reason-Based Logic as RBL rules.

We show that if the first two sentences are assumed to be true, the truth of the third sentence follows. We refer to the relations between facts, such as EXCLUSION and WEIGHING, as discussed in chapter 2, section 5. Instead of using the nonmonotonic rules of inference (chapter 2, section 6) and the corresponding technicalities of extensions, we make some 'normality assumptions', such as that a rule is not excluded.

First we note that the condition of the RBL rule concerning driving with alcohol is satisfied because Driving_with_alcohol(john) is assumed to be true.[8]

The first normality assumption is that the rule is not excluded:

```
¬Excluded(rule(driving_with_alcohol(person),
        should_be_fined(person)),
    driving_with_alcohol(john),
    should_be_fined(john))
```

Since there are no facts that lead to the exclusion of the rule, this assumption is reasonable.

---

7 Other formalizations are possible. The translation from natural to formal language is a problem that we do not discuss here.

8 Recall the convention on the translation from formulas to terms (chapter 2, section 4.3).

Using this assumption, the rule is applicable because of the relation between facts called EXCLUSION:

Applicable(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john))

APPLICABILITY makes that the fact that satisfies the condition of the rule is a reason for the rule's application:

Reason(driving_with_alcohol(john),
applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john)))

In order to use WEIGHING to conclude that the rule applies, we have to make another normality assumption, namely that there is no reason against the application of the rule:[9]

¬∃*fact_against_application*:
Reason(*fact_against_application*,
¬applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john)))

By WEIGHING_AXIOMS we have

Outweighs({driving_with_alcohol(john)},
{ },
applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john)))

9 The appearance of the following sentence may suggest that the quantification over the variable *fact_against_application* is only over a specific part of the domain, namely only over those terms that correspond to facts. However, the quantification is over the whole domain. By the definition of a language of Reason-Based Logic (chapter 2, section 4), a similar effect is obtained: a language contains no sentences of the form Reason(*fact*, *state_of_affairs*) in which *Fact* does not correspond to an instance of the condition of some rule rule(*condition*, *conclusion*).

and therefore also, by WEIGHING,

Applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john))

Using APPLICATION, the rule concerning driving with alcohol now gives,

Reason(driving_with_alcohol(john),
should_be_fined(john))

We have to make a third normality assumption, namely that there are no reasons against Should_be_fined(john):

¬∃*fact_against_fining*:
Reason(*fact_against_fining*,
¬should_be_fined(john))

Using WEIGHING_AXIOMS and WEIGHING a second time we find that

Outweighs({driving_with_alcohol(john)},
{ },
should_be_fined(john))

and finally that

Should_be_fined(john)

are true.

At three steps in the discussion above, we had to make a normality assumption. In summary, we assumed that

- The rule is not excluded.
- There is no reason against application of the rule.
- There is no reason against fining John.

These assumptions can be avoided using the nonmonotonic inference rules of Reason-Based Logic discussed in chapter 2, section 6. It can be shown that the theory consisting of the sentences

Valid(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)))
Driving_with_alcohol(john)

has a unique extension that contains

Should_be_fined(john)

and does not contain sentences contradicting the assumptions. The extension is the closure under RBL-deduction (definition 7 in chapter 2, section 6) of the set that consists of the following sentences:

Valid(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)))
Driving_with_alcohol(john)
Applicable(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john))
Reason(driving_with_alcohol(john),
applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john)))
Outweighs({driving_with_alcohol(john)},
{ },
applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john)))
Applies(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)),
driving_with_alcohol(john),
should_be_fined(john))
Reason(driving_with_alcohol(john),
should_be_fined(john))
Outweighs({driving_with_alcohol(john)},
{ },
should_be_fined(john))
Should_be_fined(john)

In the remainder of this chapter, we do not mention normality assumptions or extensions.

4 Weighing reasons in Reason-Based Logic

In this section we describe an example of weighing reasons in Reason-Based Logic in detail. We assume that the following sentences are true:

Robbing someone should be punished.
John has robbed Peter.

If we interpret the first sentence as a principle, we obtain a reason why John should be punished. Since there are no other reasons, it follows that John should be punished.

Now assume that the following sentences are also true:

Minor first offenders should not be punished.[10]
John is a minor first offender.

We find a second reason relevant concerning punishing John, but this time a reason against the fact that John should be punished.

So, there is a conflict of reasons. Without further information, Reason-Based Logic does not enforce the conclusion that John should be punished or that he should not. Both are possible. Only if it is true that one of the reasons outweighs the other, a conclusion follows.

We assume that the reason that John is a minor first offender outweighs the reason that John has robbed Peter:

'John is a minor first offender' as a reason for not punishing John outweighs the reason 'John has robbed Peter'.

In Reason-Based Logic this can be represented as follows:

Valid(rule(robbed(*person1*, *person2*),
    should_be_punished(*person1*)))[11]
Robbed(john, peter)

---

[10] In the natural language version of this sentence it is ambiguous what the scope of 'not' is. As the formal version shows, we mean 'It should not be the case that minor first offenders are punished', and not 'It should be the case that minor first offenders are not punished'.

[11] A representation of the condition of this rule that is somewhat closer to its natural language counterpart would be ∃*person2*: Robbed(*person1*, *person2*). However for simplicity the definition of the language of RBL (chapter 2, section 4) prohibits quantifiers in the conditions of rules. The condition of the rule without the existential quantifier as it is used here leads to similar consequences as the condition with the quantifier, since it can only be fulfilled if the variable *person2* is instantiated.

Valid(rule(is_minor_first_offender(*person*),
¬should_be_punished(*person*)))
Is_minor_first_offender(john)
Outweighs({is_minor_first_offender(*person1*)},
{robbed(*person1*, *person2*)},
¬should_be_punished(*person1*))

Using similar normality assumptions as in the example of section 3, it can be shown that both rules apply:

Applies(rule(robbed(*person1*, *person2*),
should_be_punished(*person1*)),
robbed(john, peter),
should_be_punished(john))
Applies(rule(is_minor_first_offender(*person*),
¬should_be_punished(*person*)),
is_minor_first_offender(john),
¬should_be_punished(john))

Applying the two rules leads to two reasons, one for and one against punishing John:

Reason(robbed(john, peter),
should_be_punished(john))
Reason(is_minor_first_offender(john),
¬should_be_punished(john))

Assuming that there are no other relevant reasons for punishing John, and using the information about the relative weight of the reasons, the relation between facts WEIGHING gives:

¬Should_be_punished(john)

It can be the case that additional reasons give rise to another conclusion. We will discuss what can happen if there is a second reason for punishing John. We add the following facts:

Injuring someone should be punished.
John has injured Peter.

These can be represented as:

Valid(rule(injury,
injured(*person1*, *person2*),
should_be_punished(*person1*)))
Injured(john, peter)

Now a second reason for punishing John arises:

Reason(injured(john, peter),
should_be_punished(john))

As a result, we cannot make the assumption that there are no other reasons for punishing John than Robbed(john, peter).

Right now, WEIGHING cannot be used to conclude whether John has to be punished or not, since there is no information about how the reasons are to be weighed.

It is possible that the additional reason does not change the result of weighing: the reason against punishing outweighs the two reasons for punishing. It should be noted that to reach a conclusion it does not suffice that the reason against punishing outweighs each of the two for punishing on its own. In that case,

Outweighs({is_minor_first_offender(*person1*)},
{injured(*person1*, *person2*)},
¬should_be_punished(*person1*))

is also true, but WEIGHING can still not be used: that would require weighing information about all three reasons together. In order to use WEIGHING it is required that

Outweighs({is_minor_first_offender(*person1*)},
{robbed(*person1*, *person2*), injured(*person1*, *person2*)},
¬should_be_punished(*person1*))

is true. In that case the conclusion that John should not be punished follows (using an appropriate normality assumption).

An interesting case, characteristic for reasoning with reasons, occurs if the two reasons for punishing John together outweigh the reason against punishing him:

Outweighs({robbed(*person1*, *person2*), injured(*person1*, *person2*)},
{is_minor_first_offender(*person1*)},
should_be_punished(*person1*))

In this case, WEIGHING leads to the opposite conclusion, viz. that John should be punished:

Should_be_punished(john)

The two pros can together outweigh the con, even if each pro on its own is outweighed by the con. This phenomenon has been called *accrual of reasons*.[12]

5 Exceptions in Reason-Based Logic

In this section, we show how exceptions can be modeled in Reason-Based Logic. We say that there is an *exception* to a rule (or principle), if the rule's (or principle's) condition is satisfied while its conclusion does not hold. It can be the case that there is another rule/principle the conclusion of which is incompatible with the conclusion of the rule/principle under consideration. In that case we speak of a conflict of rules/principles.[13] Conflicts of rules/principles are discussed in the next section.

In Reason-Based Logic, there are two main mechanisms to model exceptions to a rule/principle, namely by exclusionary reasons and by reasons against the application of a rule.[14] We discuss these in the following two subsections.

5.1 Exceptions and exclusionary reasons

Legal rules often, if not always, have scope restrictions that are not explicitly mentioned in the rule itself. For instance, in the legal rule that we already encountered about driving with alcohol,

> A person driving a car after drinking too much alcohol should be fined a considerable amount of money.

it is not explicitly mentioned in which country the rule is valid. It may be objected that this is due to the particular formulation chosen here, but also in the literal wordings in a statute the country will normally not be mentioned at all, or only in a separate section, where it is stated that the articles in the statute are only valid in a particular country.

In Reason-Based Logic, exclusionary reasons can be used to model implicit scope restrictions. For instance,

---

12 Pollock (1991a, p. 51) uses this term. He writes that it is a natural supposition that reasons accrue, but then surprisingly rejects it. We come back to Pollock's opinion in chapter 6, section 2.

13 Cf. the distinction between undercutting and rebutting exceptions (Pollock, 1987-1995): in both cases there is an exception to a rule/principle, but in case of a rebutting exception there is also a conflict of rules/principles.

14 The first mechanism has counterparts in many logical formalisms (cf. Prakken's (1993b, p. 84ff.) overview of exceptions), the second is typical for Reason-Based Logic.

Valid(rule(in_country(*country*) ∧ ¬*country* = holland,
excluded(rule(driving_with_alcohol(*person*),
should_be_fined(*person*)))))[15]

will have the effect that if John was driving in Germany, represented as

In_country(germany),

the rule concerning driving with alcohol is excluded. As a result, the rule concerning driving with alcohol is not applicable, does not apply, and does not lead to the conclusion that John should be fined. (Of course, it is possible that the same conclusion nevertheless follows due to another valid rule, e.g., a German rule of law, that is not excluded.)

Scope restrictions for a class of rules can be represented by explicit knowledge on the origin of the rules. For instance, using the explicit knowledge on which articles rules are based and which articles are in the penal code, all rules that are based on articles in the penal code are restricted to Holland by the following:

Valid(rule(in_country(*country*) ∧ ¬ *country* = holland
∧ based_on(*rule*, *article*) ∧ in_penal_code(*article*),
excluded(*rule*)))

An obvious objection to this type of representation of exceptions, viz. *outside* the rule, is that since they are often explicitly available they can be made part of the rule during the translation of the legal rule to its formal counterpart. For instance, this would lead to the following representation:

Valid(rule(driving_with_alcohol(*person*) ∧ in_country(holland),
should_be_fined(*person*)))

There are drawbacks to this approach, as is generally accepted (see chapter 4). First, it can easily lead to very long rule conditions, most of which have to be repeated in many rules and are almost always unimportant for a particular case. For analogous reasons, in actual codifications of legal rules scope restrictions are not explicitly stated in each rule. Second, the dissimilarity in structure of the informal and the formal representation is unnecessarily enlarged.[16] As a result, translation in either direction becomes harder, which is particularly a problem in a constantly

[15] We have made a simplification here, since facts are often dependent on a situation or case. For instance, a rule can apply to *a case*. As a result, many predicates would need an extra variable for cases. For convenience we leave cases implicit. For instance, in the following In_country(*country*) means that the case *at hand* is in the country represented by *country*.

[16] Cf. the desirability of an *isomorphic* representation of the law (see e.g. Bench-Capon and Coenen, 1992).

changing domain, such as the law. Third - and this is a drawback that cannot be overcome - not all exceptions to legal rules are explicitly available since it is impossible to anticipate all cases in which a rule is not applicable.

The third point brings us to the second way of representing exceptions in Reason-Based Logic.

5.2 Exceptions and reasons against application

When a legal rule is made by the legislator, not all cases that fall inside the definition set by a legal rule can be foreseen, if not fundamentally, then at least in practice. We do not treat the philosophical side of these matters, but give a concrete example.

It can happen that there is a case that falls within the rule's condition and to which the rule is applicable, but to which the rule should not apply for some other reason. For instance application of the rule might be against its purpose.

We assume that there is a rule that forbids sleeping in the railway station. The rule has as its purpose to prevent tramps from occupying the station as their place to spend the night. An old lady that wants to meet a friend at the station dozes off when the evening train turns out to be late. Should the prohibition apply to this lady?[17]

The following two sentences describe the case:

Valid(rule(sleep_in_station(*act*),
    forbidden(*act*)))
Sleep_in_station(lady's_act)

We assume that application of the rule about the sleeping prohibition in the case of the lady is against the rule's purpose:

Application_against_purpose(rule(sleep_in_station(*act*),
        forbidden(*act*)),
    sleep_in_station(lady's_act),
    forbidden(lady's_act))

Hence, we need a general rule stating that if application is against the purpose of a rule, this is a reason not to apply the rule:

Valid(rule(application_against_purpose(*rule*, *fact*, *state_of_affairs*),
    ¬applies(*rule*, *fact*, *state_of_affairs*)))

[17] This example is inspired by Fuller's (1958, p. 664). The formulation here is taken from Hage and Verheij (1994a, b).

Since the condition of the rule about the sleeping prohibition is satisfied, we have a reason to apply it (by APPLICABILITY):

Reason(sleep_in_station(lady's_act),
applies(rule(sleep_in_station(*act*),
forbidden(*act*)),
sleep_in_station(lady's_act),
forbidden(lady's_act)))

But we also have a reason against application:

Reason(application_against_purpose(rule(sleep_in_station(*act*),
forbidden(*act*)),
sleep_in_station(lady's_act),
forbidden(lady's_act)),
¬applies(rule(sleep_in_station(*act*),
forbidden(*act*)),
sleep_in_station(lady's_act),
forbidden(lady's_act)))

We suppose that the reason against application of the rule because of its purpose outweighs the reason for application because of the applicability of the rule:

Outweighs({application_against_purpose(rule(sleep_in_station(*act*),
forbidden(*act*)),
sleep_in_station(lady's_act),
forbidden(lady's_act))},
{sleep_in_station(lady's_act)},
¬applies(rule(sleep_in_station(*act*),
forbidden(*act*)),
sleep_in_station(lady's_act),
forbidden(lady's_act))

We now conclude

¬Applies(rule(sleep_in_station(*act*),
forbidden(*act*)),
sleep_in_station(lady's_act),
forbidden(lady's_act))

Because the rule about the sleeping prohibition is not applied, it does not lead to the prohibition of the lady's sleeping.

6 Conflicts in Reason-Based Logic

We speak of a conflict of rules/principles, if there is a group of rules/principles the conclusions of which are incompatible, while their conditions are satisfied. There are two main mechanisms in Reason-Based Logic to deal with conflicts of rules and principles, namely by means of exclusionary reasons and by means of weighing reasons.

6.1 Conflicts and exclusionary reasons

When dealing with conflicting legal rules, several types of so-called *conflict rules* are used in law: specific priority clauses for pairs of rules, or for classes of rules, and general rules such as Lex Superior, Lex Posterior, and Lex Specialis. The effect is that one or more of the conflicting rules are excluded and that in the end there is no conflict left.

Such conflict rules can be represented in Reason-Based Logic by means of exclusionary reasons. For instance, following Prakken (1993b), if there is a contract with features of lease of business accommodation and of another type of contract, and there is a conflict between a legal rule dealing with such lease contracts and one dealing with contracts of the other type, the first rule prevails according to Section 7A: 1624 of the Dutch civil code. This legal rule might be represented as follows:

Valid(rule(deals_with_lease_of business_accommodation(*rule1*)
∧ applies(*rule1*)
∧ deals_with_contracts_of_another_type(*rule2*)
∧ in_conflict(*rule1*, *rule2*),
excluded(*rule2*)))

More generally, explicit knowledge about prevalence can be used, for instance:

Valid(rule(applies(*rule1*)
∧ in_conflict(*rule1*, *rule2*)
∧ prevails_over(*rule1*, *rule2*),
excluded(*rule2*)))

Using the latter rule about prevalence, a conflict rule such as Lex Posterior can be represented as follows:

Valid(rule(more_recent(*rule1*, *rule2*),
prevails_over(*rule1*, *rule2*)))

It has to be specified when rules are in conflict. It can for instance be specified that rules are in conflict when they have opposite conclusions.[18]

In practice, it can happen that conflict rules are themselves involved in a conflict. For instance, a rule can be of earlier date and of higher authority than another rule. Since the conflict rules are themselves represented as rules in Reason-Based Logic, such conflicts of conflict rules can be approached in the same way as conflicts in general.

6.2 Conflicts and weighing reasons

The second mechanism to deal with conflicting rules/principles is by the weighing of the resulting reasons.[19] An example was already discussed in section 4 of this chapter.

Not all rules and principles involved in a conflict lead to conflicting reasons, since there can be rules/principles that do not apply because of exclusionary reasons, or reasons against their application. If after such simplifications of the conflict there is still a conflict of reasons, information about their relative weight can resolve the conflict and lead to a final conclusion. So, there are several layers in which a conflict of rules/principles is simplified before the resulting reasons are weighed. Figure 7 gives an overview.

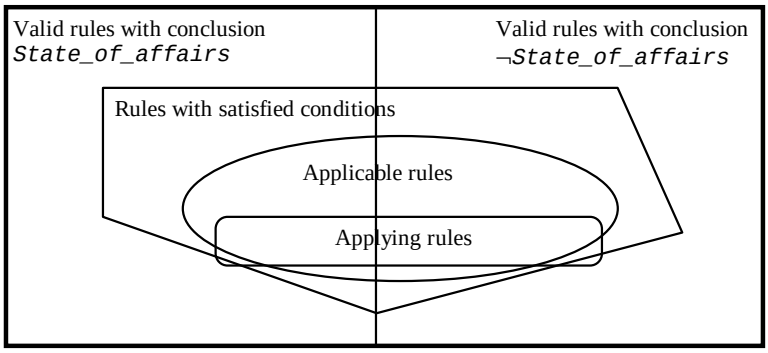

Figure 7: Not all conflicting RBL rules lead to conflicting reasons.

It may seem strange that the applying RBL rules are not indicated as a subset of the applicable rules. In section 8.3 on the analogous application of a rule, we will see an example of an RBL rule that applies, while it is not applicable.

---

[18] The need for specifying when rules are in conflict can be considered a drawback since it puts a heavy burden on the domain theory. However, it can also be considered an advantage since it can make the notion of conflict more manageable.

[19] This mechanism can only deal with conflicts of rules/principles with opposite conclusions, due to the notion of weighing as modeled in Reason-Based Logic.

As a final remark about dealing with conflicts of rules/principles in Reason-Based Logic, we stress that Reason-Based Logic does not resolve all conflicts, and merely provides different means to represent conflict-resolving information. For instance, the following set of sentences does not have an extension in Reason-Based Logic due to an unresolved conflict of rules:

A
B
Valid(rule(a, c))
Valid(rule(b, d))
$\neg C \vee \neg D$

However, there is no inconsistency (in the sense of RBL-deduction), and the conflict is resolved if the sentence Excluded(rule(a, c), a, c) or the sentence Excluded(rule(b, d), b, d) is added.

7 Rules and principles in Reason-Based Logic

We now return to our integrated view on rules and principles, as introduced in section 2. Recall that our view was based on two assumptions:

- Both rules and principles give rise to reasons if they are applied.
- The differences between reasoning with rules and principles result from different types of relationships with other rules and principles, which may interfere.

In section 7.1, we discuss our basic example of the role of the relationships between rules and principles, namely a rule with underlying principles. In section 7.2, we return to the differences between rules and principles as discussed in section 1.

7.1 A rule and its underlying principles

In section 2.1, we discussed the Dutch legal rule of Art. 7A:1612 BW that sale of a house should not terminate an existing rent contract. This rule can be represented in Reason-Based Logic as follows:

Valid(rule(sale_house,
    ought_to_be_done(continuation_contract)))

We considered two principles underlying this rule, namely a pro-principle that somebody who lives in a house should be protected against measures that threaten

the enjoyment of the house, and a con-principle that contracts only bind the contracting parties. These principles can be represented as RBL rules as follows:

Valid(rule(protects_inhabitants(*act*),
    ought_to_be_done(*act*)))
Valid(rule(¬party_bound_by_contract,
    ¬ought_to_be_done(continuation_contract)))

The fact that these principles underlie the rule of Art. 7A:1612 BW is represented as:

Underlies(rule(protects_inhabitants(*act*),
        ought_to_be_done(*act*)),
    rule(sale_house,
        ought_to_be_done(continuation_contract)))
Underlies(rule(¬party_bound_by_contract,
        ¬ought_to_be_done(continuation_contract)),
    rule(sale_house,
        ought_to_be_done(continuation_contract)))

The rule and its underlying principles are schematically shown in Figure 8.

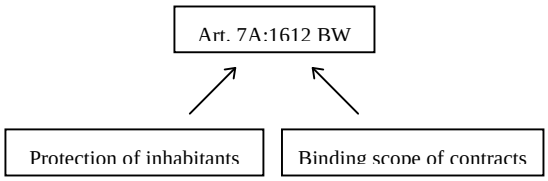

Figure 8: The rule of Art. 7A:1612 BW and its underlying principles

If a house with renting inhabitants is sold, the two principles lead to conflicting reasons, since continuation of an existing rent contract protects the inhabitants of a house, while the new owner is not bound by the contract with the inhabitants. We have

Protects_inhabitants(continuation_contract)
¬Party_bound_by_contract

and therefore the two RBL rules about the protection of inhabitants and about the binding scope of contracts lead to the conflicting reasons:

Reason(protects_inhabitants(continuation_contract),
    ought_to_be_done(continuation_contract))

Reason(¬party_bound_by_contract,
¬ought_to_be_done(continuation_contract))

However, by making the legal rule of Art. 7A:1612 BW, the legislator has balanced the conflicting principles, and decided how the reasons generated by them should be weighed against each other. Therefore, if we have the fact

Sale_house

the rule of Art. 7A:1612 BW should lead to the conclusion

Ought_to_be_done(continuation_contract)

without the interference of the two underlying principles: the rule of Art. 7A:1612 BW replaces its underlying principles if it applies (see section 2.1), and the two principles should not be applicable. The required situation is shown in Figure 9.

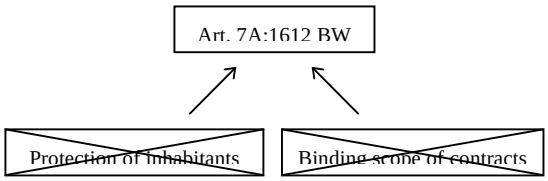

Figure 9: The rule of Art. 7A:1612 BW replaces its underlying principles if it applies

In Reason-Based Logic, replacement can be modeled using exclusionary reasons. We need the following rule:

Valid(rule(underlies(*rule1*, *rule2*) ∧ applies(*rule2*),
excluded(*rule1*)))[20]

Since we can conclude

Applies(rule(sale_house,
ought_to_be_done(continuation_contract)),
sale_house,
ought_to_be_done(continuation_contract))

[20] Henry Prakken has correctly noted that *rule2* also excludes *rule1* in case there is another rule or principle that does not underlie *rule2*, but nevertheless interferes. As a result, there can be no interaction of the other rule or principle with *rule1* if *rule2* applies. This does not always seem desirable, and deserves further study. Interestingly, in this situation *rule2* is not a typical rule.

we find:

Excluded(rule(protects_inhabitants(*act*),
ought_to_be_done(*act*)),
protects_inhabitants(continuation_contract),
ought_to_be_done(continuation_contract))
Excluded(rule(¬party_bound_by_contract,
¬ought_to_be_done(continuation_contract)),
¬party_bound_by_contract,
¬ought_to_be_done(continuation_contract))

The principles about the protection of inhabitants and about the binding scope of contracts do no longer lead to reasons. As a result, the rule of Art. 7A:1612 BW leads without interference to the conclusion

Ought_to_be_done(continuation_contract),

just as required.

7.2 The differences between rules and principles

We can now finish our integrated view on rules and principles as represented in Reason-Based Logic. As in the case of a rule that replaces its underlying principle, a typical rule is an RBL rule that leads to exclusionary reasons against the applicability of any interfering rule or principle. A typical principle is an RBL rule that does not exclude any interfering rule/principle. Interfering rules and principles are typically rules and principles with equal or opposite conclusion.

This is in line with our two main assumptions:

- Both rules and principles give rise to reasons if they are applied. The difference between the two is that an applying rule not only generates a reason for its conclusion, but also exclusionary reasons for the principles it replaces.
- The differences between reasoning with rules and reasoning with principles result from different types of relationships with other rules and principles, interfering with them: rules lead to exclusionary reasons to interfering rules and principles, while principles lead to reasons that are weighed in case of a conflict.

It is clear that in this view there is no clear border between rules and principles. For instance, an isolated rule cannot be distinguished from an isolated principle. Only if there are interfering rules and principles, gradual differences can be seen. On the one extreme there is the typical principle that, if it applies, does not generate exclusionary reasons for any of the rules and principles that interfere with it. On the other extreme there is the typical rule that, if it applies, excludes all interfering rules

and principles. In between the two extremes there are many degrees of hybrid rules/principles, some more principle-like, others more rule-like.

In section 1, we discussed three differences between rules and principles. First, it seemed that rules lead directly to their conclusion if they apply, while principles lead to reasons that have to be weighed. This difference has disappeared since in our view both rules and principles generate reasons. Therefore both rules and principles first lead to reasons that are then weighed. Nevertheless, also in our view, rules *seem* to lead directly to their conclusion. This is the result of the fact that in the case of an applying rule no weighing of reasons is necessary since all interfering rules and principles are excluded. Therefore, the step from reason to conclusion seems immediate.

Second, it seemed that conflicting rules lead to a contradiction if they apply, while conflicting principles merely lead to conflicting reasons. In our representation, no real contradiction can arise by the application of rules with opposite conclusions, since rules just as principles only generate reasons. Moreover if an apparent rule gives rise to a reason that conflicts with another reason, this is a sign that it is *not* a typical rule, but has a somewhat more principle-like character.

Third, it seemed that rules lead to their conclusion in isolation, while principles interact with other principles: additional relevant reasons arising from other principles can influence the result of weighing. In our view, this seeming difference is beside the point since rules in isolation do not differ from principles in isolation. The rule-like character of a rule can only be appreciated if there are interfering rules or principles.

8 Analogy in Reason-Based Logic

The last topic that we discuss is reasoning by analogy.[21] As an application of our integrated view on rules and principles, we describe three different ways of reconstructing reasoning by analogy. To avoid misunderstanding, we stress that our approach to reasoning by analogy is not based on cases,[22] but on rules and principles. Instead of using the similarity and dissimilarity of cases as criteria to justify reasoning by analogy, we use the relationships between rules and principles.

We assume that in reasoning by analogy there is a rule that does not apply because its condition is not satisfied, but that nevertheless its conclusion holds on the basis of additional information about the relationships between the rule and other rules and principles. We distinguish three forms of reasoning to analyze reasoning by analogy:

- Application of principles that underlie the original rule that does not apply itself.

21 This section is based on Verheij and Hage (1994).
22 See, for instance, Ashley (1990), Yoshino *et al.* (1993) and Tiscornia (1994).

- Application of an analogous rule/principle that has the same underlying principles as the original rule that does not apply.
- Analogous application of the original rule, i.e., application of the rule with a 'non-standard' justification, based on, for instance, a principle.

We do not claim that these three forms of reasoning are always cases of reasoning by analogy, but that they are useful means to analyze a given case of reasoning by analogy. Below we use one example, and analyze it by the three mentioned forms of reasoning.

8.1 Application of underlying principles

In the first form of reasoning by analogy, the principles apply that underlie the original rule that does not apply itself.

The example we use is based on Art. 7A:1612 BW. It was also used in the sections 2.1 and 7.1 to explain the replacement of the principles underlying a rule. Again, we have one rule and two underlying principles:

Valid(rule(sale_house,
    ought_to_be_done(continuation_contract)))
Valid(rule(protects_inhabitants(*act*),
    ought_to_be_done(*act*)))
Valid(rule(¬party_bound_by_contract,
    ¬ought_to_be_done(continuation_contract)))
Underlies(rule(protects_inhabitants(*act*),
        ought_to_be_done(*act*)),
    rule(sale_house,
        ought_to_be_done(continuation_contract)))
Underlies(rule(¬party_bound_by_contract,
        ¬ought_to_be_done(continuation_contract)),
    rule(sale_house,
        ought_to_be_done(continuation_contract)))

Here we assume that a house with renting inhabitants is not sold, but donated. So, we have the facts:

¬Sale_house
Donation_house

As a result, the condition of the rule of Art. 7A:1612 BW is not satisfied, and the rule does not apply. But just as in the case of sale, continuation of the existing rent contract is a way to protect the inhabitants, while the new owner is not bound by the existing contract:

Protects_inhabitants(continuation_contract)
¬Party_bound_by_contract

Therefore, the conditions of the principles about the protection of inhabitants and about the binding scope of contracts are satisfied. Since the rule of Art. 7A:1612 BW does not apply, the replacement rule

Valid(rule(underlies(*rule1*, *rule2*) ∧ applies(*rule2*),
excluded(*rule1*)))

does not give exclusionary reasons for the two underlying principles. They apply and give rise to the reasons:

Reason(protects_inhabitants(continuation_contract),
ought_to_be_done(continuation_contract))
Reason(¬party_bound_by_contract,
¬ought_to_be_done(continuation_contract))

The situation is shown in Figure 10.

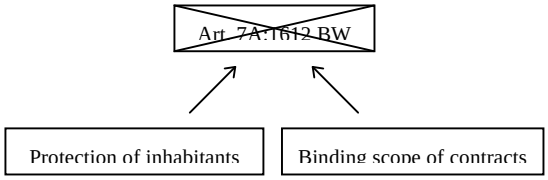

Figure 10: The principles underlying the rule of Art. 7A:1612 BW apply

So, in the case of donation two reasons arise that are based on the same principles as those taken into account by the legislator, when the original rule was made.

There are good reasons to assume that the weighing of these reasons has the same outcome as in the reasoning of the legislator:

Outweighs({protects_inhabitants(continuation_contract)},
{¬party_bound_by_contract},
ought_to_be_done(continuation_contract))

and leads to the same conclusion that the contract should be continued:

Ought_to_be_done(continuation_contract)

In this analysis, two principles applied in the case of donation. They are precisely the two principles that were replaced in the case of sale. The case of donation is therefore in a sense of *the same kind* as the case of sale. Therefore we speak of a

form of reasoning by analogy. If only some of the underlying principles apply, or more goals and principles are relevant, we cannot always speak of a case of reasoning by analogy. The case might even be solved differently, since the reasons might be weighed differently.

8.2 Application of an analogous rule/principle

In the second form of reasoning by analogy, a analogous rule/principle applies that has the same underlying principles as the original rule. This leads to another analysis of the same example.

In our example the analogous rule/principle might be:

```
Valid(rule(donation_house,
    ought_to_be_done(continuation_contract)))
```

The legal decision maker that wants to base his reasoning on this rule has to justify its validity. This justification can be based on the same reasons as the ones used by the legislator when he made Art. 7A:1612 BW:

```
Reason(protects_inhabitants(continuation_contract),
    valid(rule(donation_house,
        ought_to_be_done(continuation_contract))))
Reason(¬party_bound_by_contract,
    ¬valid(rule(donation_house,
        ought_to_be_done(continuation_contract))))
```

In this line of reasoning, the two reasons are not relevant for the conclusion that the contract should be continued, but for the validity of the new RBL rule about donation. In their new role, the reasons might be weighed the same way as before:

```
Outweighs({protects_inhabitants(continuation_contract)},
    {¬party_bound_by_contract},
    valid(rule(donation_house,
        ought_to_be_done(continuation_contract)))
```

The conclusion is that the RBL rule about donation is valid.

It may seem that there is a problem here with the separation of powers: while the legislator can make rules, the legal decision maker cannot. However, this problem is only seeming, and due to the different meanings of rule validity in law and in reasoning. We use the term ‘rule validity’ in the latter sense. For rule validity in that sense the separation of powers is irrelevant.[23]

---

[23] In Verheij and Hage (1994), we put it differently: we wrote that the legal decision maker can only validate legal principles (and not legal rules) because of the separation of powers.

If the rule about donation applies, the principles about the protection of inhabitants and about the binding scope of contracts are again replaced by the rule about donation and do not apply. An overview of the relations of the rules and principles involved in this reasoning is shown in Figure 11.

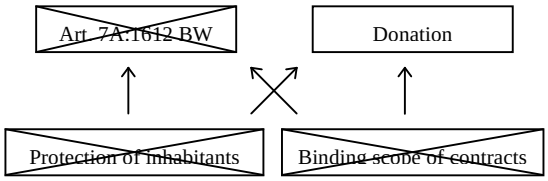

Figure 11: The rule about donation applies having the same underlying principles as the original rule of Art. 7A:1612 BW

Since the rule about donation has the same underlying principles as the rule of Art. 7A:1612 BW we say that a rule is applied analogous to the original rule.

8.3 Analogous application of the original rule

The third form of reasoning by analogy is typical for Reason-Based Logic, since it involves reasons for and against applying a rule.

In this third analysis of the example, the rule of Art. 7A:1612 BW is not applicable, since its condition is not satisfied, just as in the previous two analyses. As a result, the standard reason for applying the rule, based on the relation between facts APPLICABILITY (chapter 2, section 5), does not arise. However, a rule that is not applicable can apply, since there can be other reasons that lead to its application.

In our case, the reasons are again those for and against the continuation of the contract having a new role. They now are represented as follows:

```
Reason(protects_inhabitants(continuation_contract),
    applies(rule(sale_house,
            ought_to_be_done(continuation_contract)),
        sale_house,
        ought_to_be_done(continuation_contract)))
Reason(¬party_bound_by_contract,
    ¬applies(rule(sale_house,
            ought_to_be_done(continuation_contract)),
```

However, in the line of reasoning described in the text the two underlying principles are replaced if the RBL rule about donation applies. Otherwise the reasons arising from these principles would be taken into account twice. As a result, the RBL rule about donation has a rule-like character.

sale_house,
ought_to_be_done(continuation_contract)))

Here the reasons protects_inhabitants(continuation_contract) and ¬party_bound_by_contract are reasons for and against applying the rule of Art. 7A:1612 BW, respectively. Again the result of weighing these reasons might be the same in this new role, as in section 8.2:

Outweighs({protects_inhabitants(continuation_contract)},
{¬party_bound_by_contract},
applies(rule(sale_house,
ought_to_be_done(continuation_contract)),
sale_house,
ought_to_be_done(continuation_contract)))

As a result, we can conclude that the rule of Art. 7A:1612 BW applies, even though its condition is not satisfied and it is not applicable:

Applies(rule(sale_house,
ought_to_be_done(continuation_contract)),
sale_house,
ought_to_be_done(continuation_contract))

Since the rule of Art. 7A:1612 BW applies, it replaces its underlying principles by the replacement rule, just as any applying rule: the principles about the protection of inhabitants and about the binding scope of contracts are excluded and do not apply. Figure 9 shows the relations of the rules and principles involved (but does not show the reasons in their new role). These relations are the same as in the case of normal rule application. Since in this example the rule does apply, but not for the standard reason that its condition is satisfied, we call this *analogous* rule application.