CONSISTENCY OF RULES AND NORMS[1]

JAAP HAGE

Maastricht University

Department of Metajuridica

P.O Box 616

6200 MD Maastricht

email: jaap.hage@metajur.unimaas.rl

version: 02/11/13

Abstract

This paper develops the theory that a set of rules is consistent if it is not possible that 1) the conditions of the rules in the set are all satisfied, 2) there is no exception to either one of the rules, and 3) the consequences of the rules are incompatible. To this purpose the notion of consistency is generalised to make it cover rules and is relativised to a background of constraints.

It is argued that a similar theory is also useful to characterise the consistency of deontic sentences of the ought-to-do type.

The theory about rule consistency is formalised by means of Rule Logic, in which rules are treated as constraints on the possible worlds in which they exist. Rule Logic itself is introduced by giving a model-theory for it. It is characterised by means of constraints on worlds that are possible according to Rule Logic. The formal theory is refined by disallowing ungrounded exceptions to rules. To that purpose an additional constraint is imposed on worlds that are possible according to Rule Logic.

1 Introduction and overview

A legal system consists of - amongst others - a set of rules. A legislator would like the legislation of a particular legal system to be consistent, in the sense that it does not lead to incompatible outcomes for a specific case. The notion of consistency of rules is of practical interest for the development of legal systems.

The purpose of this paper is to introduce and to develop a theory about the consistency of rules. There are at least three reasons why the consistency of rules differs from the consistency of descriptive sentences, all of which will be discussed in the following sections. First, many rules have a conditional structure, but their consistency cannot be treated as the consistency of conditional sentences. Second, the consistency of rules is relative to a set of constraints that determine which states of affairs can go together. Part of the complications in connection with rule consistency is that rules themselves can function as constraints relative to which consistency has to be judged. And finally, reasoning with

[1] This paper is a substantially expanded version of Hage 1999 and Hage forthcoming.

rules is defeasible. There can be exceptions to rules that block the application of rules the conditions of which are satisfied. These exceptions can prevent threatening rule conflicts, thereby making seemingly inconsistent rules consistent. I will try to develop a theory of rule consistency that takes all these three aspects into account.

My topic is easily confused with another one that has been discussed quite often in the literature. I refer to *deontic consistency*, also called normative consistency.[2] This concerns questions as whether there *can* be logical relations between deontic sentences, prescriptions, or norms[3] such as that it is forbidden to steal and that it is permitted to steal.[4] Deontic consistency is related to the issue that I will be dealing with, but it is not the same one. The rules that can be (in)consistent need not be deontic at all. The question whether the conceptual rules (legal definitions) that surf boards count as vehicles and that nothing without wheels counts as a vehicle, are consistent falls, for instance, under the topic of this paper, but has nothing to do with deontic consistency. In section 7 I will apply some of the insights derived from the first six sections to characterise the consistency of deontic sentences of the ought-to-do type.

This paper is divided into two parts. In the first part, consisting of the sections 2 to 7, the theory of rule consistency is elaborated in an informal way. In the second part, consisting of the sections 7 to 11, the theory is developed formally. Section 2 deals with the complications raised by the conditional structure of rules for the notion of rule consistency. The discussion of this section leads to a first preliminary formulation of the theory of rule consistency. In section 3 this formulation is refined on the basis of a discussion of the compatibility of states of affairs. This section also introduces the notion of a constraint. Section 4 elaborates the view that rules can be seen as a kind of constraints and its implications for the theory of rule consistency. Section 5 adds a little to the theory for the case of conditionless rules. In section 6 a major complication is added by allowing rules to have exceptions. Section 7 dealt with the consistency of deontic sentences.

The first step of the formalisation, taken in section 8, is the introduction of the logical language L_{RL} for Rule Logic. The second step (section 9) is to characterise Rule Logic in terms of worlds that are possible according to this logic. Section 10 formalises the notion of compatibility of states of affairs in terms of worlds that are possible according to Rule Logic. Section 11 makes use of this formalisation to give a first formal characterisation of rule consistency. This formalisation does not yet take into account that there should be as few exceptions to rules as possible. Section 12 does the same for the

2 An overview can be found in Den Haan 1996. See also Hamner Hill 1987 and Lindahl 1992.

3 The question after the precise nature of the entities between which logical relations can or cannot exist is part of the issue.

4 Deontic consistency is the issue at stake in the discussion of norm conflicts in the sense of Kelsen 1979, of norm contradictions and norm collisions in the sense of Hamner Hill 1987, of norm consistency in the sense of Von Wright 1991, of disaffirmation conflicts and compliance conflicts in the sense of Lindahl 1992, and of norm conflicts in the sense of Ruiter 1997.

consistency of deontic sentences. Section 13 adds a constraint to the characterisation of possible worlds in order to guarantee that there are no 'free floating' exceptions.

PART 1

2 Rules as conditionals

If the consistency of rules were the same as that of conditional sentences, the following two rules would be consistent:

Thieves are punishable.
Thieves are not punishable.

Their consistency would follow from that the following sentences are *not* inconsistent:

$\forall x(\mathtt{Thief}(x) \rightarrow \mathtt{Punishable}(x))$

$\forall x(\mathtt{Thief}(x) \rightarrow {\sim}\mathtt{Punishable}(x))$

Instead of being inconsistent, these sentences allow the derivation of

${\sim}\exists x(\mathtt{Thief}(x))$

Intuitively, however, a legislator should not be able to remove thieves from the world, merely by issuing both the rules that thieves are punishable and that they are not punishable.

The first conclusion to draw is, therefore, that a theory in which rules are treated like descriptive sentences, and that considers rule consistency as similar to the consistency of descriptive sentences, is unsatisfactory. There is reason to search for a notion of consistency that is especially suited to rules.[5] A relevant intuition in this connection is that the consistency of rules should not depend on whether certain states of affairs obtain. We want, for instance, the rules that thieves are punishable and that they are not punishable to be inconsistent independent of whether there are thieves. If there are thieves, the two rules can, barring exceptions, be used to derive an inconsistency in the traditional sense, because then it can be derived that these thieves are both punishable and not. However, we want the inconsistency of the rules to be independent of whether there are facts that satisfy their conditions.

Yet, it is important for the consistency of rules whether the rules *can* be applied to the same case. For instance, the rules that thieves are punishable and that non-thieves are not punishable are intuitively consistent. The inconsistency of rules depends both on the incompatibility of the conclusions of the rules, and on the compatibility of the rule conditions. The basic idea is that a set of rules is inconsistent

[5] Obviously, the reasons given here why rule consistency is different from the consistency of descriptive sentences are not decisive. It is well possible to treat rule consistency in the same way as consistency of descriptive sentences and take the phenomenon that rules are only inconsistent if certain facts are present, into the bargain. Readers who buy this last view should read this paper as a study of the implications of another view.

if it is possible that there is a case in which the conditions of all the rules are satisfied, while the consequences that are attached to this case by these rules are incompatible.

This basic idea needs to be refined, however. For instance, the rules that thieves are punishable and that minors are not, are inconsistent, because there is a possible case (a minor thief) to which the rules attach incompatible consequences. If the rule that non-thieves are not punishable is added to these two rules, there cannot be a case anymore that satisfies the conditions of all the three rules, because it is not possible that somebody is both a thief and a non-thief. According to the basic idea about rule consistency, the resulting set of three rules would be consistent. It should not be possible, however, to make a set of rules consistent by adding a rule which has conditions that are incompatible with the conditions of the rules in the inconsistent set.

To avoid this complication, the demand is made that a set of consistent rules cannot contain an inconsistent subset. Or, in other words, a set of rules is inconsistent if it contains an inconsistent subset.[6]

This leads me to the following provisional theory about rule consistency:

The rules in a set s are consistent if and only if it is not so that there are a subset s' of s and a possible case f such that

a. the facts in f satisfy the conditions of all the rules in s', and

b. *to which the rules in s' attach incompatible consequences.*

This provisional theory will be developed and expanded in the rest of this paper.

3 Consistency, compatibility and constraints

Descriptive sentences are called *consistent* if it is possible that they are all true. For instance, the sentences 'John is a thief' and 'John is a minor' are consistent, because it is possible that John is both a thief and a minor. In other words, because the *states of affairs* that John is a thief and that he is a minor are *compatible*, the *sentences* that express these states of affairs are *consistent*.[7] The sentences 'John is a thief' and 'John is not a thief' are inconsistent, because it is not possible that John both is and is not a thief. It is the *incompatibility* of the *states of affairs* that John is a thief and that he is not a thief that makes the corresponding *sentences inconsistent*.

Compatibility of states of affairs is always relative to some background of *constraints*.[8] The states of affairs that John is a thief and that he is not a thief are incompatible because of the constraint that a

---

6 Obviously, the subset need not be a proper one.

7 I use the notion of a state of affairs as that what is expressed (not: *denoted*) by a descriptive sentence. If a sentence is true, the state of affairs that it expresses obtains and is a *fact*. Following Strawson (1971, p. 195) I consider states of affairs to be language-dependent. The actual world is the set of all facts; a possible world is a set of states of affairs. The notion of a state of affairs is given a precise definition in section 8.1.

8 This point is, in a different context, also made by Prakken and Sartor 1996, p. 184/5.

state of affairs cannot both obtain and not obtain. A similar constraint is that the compound state of affairs that John is both a thief *and* a minor can only obtain if *both* the states of affairs that John is a thief and that he is a minor obtain. Such constraints are usually called *logical* constraints. Besides logical constraints, there are also other constraints. There are *physical* constraints that prevent somebody from being in two non-adjacent countries at the same time. It is, for instance, physically impossible that John is both in France and in Austria. *Conceptual* constraints make it impossible that something else than a point is both a square and a circle.

This is the occasion to introduce a terminological convention. The expressions 'compatible' and 'incompatible' will be used for *states of affairs* which can, or cannot go together relative to a set of constraints. The expressions 'consistent' and 'inconsistent' are used for both descriptive sentences and for rules, with different criteria for sentence and rule consistency.

What is possible depends on the constraints that are taken into account. I will develop this idea by means of the notion of a possible world. A state of affairs is possible (can obtain), if there is at least one possible world in which this state of affairs obtains. A world is possible relative to some set of constraints c, if the facts of that world satisfy the constraints in c.

The most fundamental constraints on a possible world are the logical constraints. A world that is physically possible must (also) satisfy the constraints of physics. These are the physical laws and everything that follows from them. For instance, in a physically possible world, it does not occur that there are two bodies with gravitational mass that do no exercise mutual gravitational forces. Neither does it occur in a physically possible world that something travels faster than the speed of light in vacuum.[9]

In a legally possible world it does not occur that somebody is a thief and not punishable.[10] As this last example shows, the constraints on possible worlds can be the result of human culture. By adopting rules, humans can impose additional constraints on the world in which they live. In contrast to physical constraints, rule-based constraints are contingent in the sense that they are absent in a world in which these rules do not exist. But when they exist, they rule out certain combinations of states of affairs as impossible, and necessitate other states of affairs.

If the compatibility of states of affairs is relative to a set of constraints, this has implications for our provisional definition of rule consistency:

The rules in a set s are consistent relative to a set of constraints c if and only if it is not so that there are a subset s' of s and a possible case f such that

a. *the states of affairs in f are compatible relative to c,*
b. *the states of affairs in f satisfy the conditions of all the rules in s', and*

---

9 Obviously these examples reflect both the present state of physical knowledge and my limited command of it.

10 Exceptions to rules are disregarded at this stage of the presentation.

c. *the rules in s' attach consequences to f that are incompatible relative to c.*

I will treat logical compatibility of states of affairs as a special case, that is as compatibility relative to the empty set of constraints. In the second part of this paper, where the present theory is formalised, this idea will be made more precise.

Let me illustrate the implications of the above theory of rule consistency by means of some examples. In these examples I use the formalism that will be explained in the second part of this paper. Because of its similarity to the language of predicate logic, I assume that this will cause no problems.

EXAMPLE 1

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *thief(x) ⇒ *~punishable(x)
```

The rules 1 and 2 are logically inconsistent, because if John is a thief, this fact satisfies the conditions of both rules, while the conclusions of the two are logically incompatible. Notice that the inconsistency of the rules does not depend on the presence of the fact that John is a thief. This fact merely illustrates the inconsistency.

EXAMPLE 2

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
```

The rules 1 and 2 are logically inconsistent, for the same reasons as in example 1. The inconsistency is illustrated by the case of John who is both a thief and a minor.

EXAMPLE 3A

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *protected(x)
rule_3 = *protected(x) ⇒ *~punishable(x)
```

The rules 1, 2 and 3 are logically inconsistent, because the rules 1 and 3 are logically inconsistent. Rule 2 plays no role in this connection. In example 3B in the next section we will encounter a related situation in which rule 2 has a role to play.

4 Rules as constraints

Rules function as constraints on the worlds, or on the normative systems, in which they exist. In the Netherlands the rule exists that thieves are punishable. As a consequence the states of affairs that somebody is a thief and that he is not punishable are, barring exceptions to the rule, incompatible according to Dutch law. In a legal system where this rule does not exist, these states of affairs might be compatible.

The phenomenon that rules can function as constraints on possible worlds has implications for the above theory of rule consistency. To illustrate this, I will adapt example 3A:

EXAMPLE 3B

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *protected(x)
c = {rule_3: *protected(x) ⇒ *~punishable(x)}
```

The third rule of example 3A is removed from the set of rules that is evaluated as to its consistency, and added to the set c of constraints that govern the world in which the rules 1 and 2 are evaluated. The first thing to notice is that the remaining rules 1 and 2 are *logically* consistent. This is not surprising, because the inconsistency of the rules 1 to 3 in example 3A depended on the presence of rule 3. If this rule is removed from the set, the logical consistency is restored.

However, if the removed rule is added to the constraints relative to which consistency is evaluated, the rules 1 and 2 become inconsistent relative to the constraint in c, since this constraint makes the states of affairs that somebody is punishable and that he is protected incompatible. So the rules 1 and 2 are logically consistent, but they are inconsistent relative to the set c of constraints which includes rule 3.

It makes no difference whether the rules 1 to 3 are evaluated together as to whether they are logically consistent, or that the consistency of the rules 1 and 2 is evaluated against the background of rule 3. Nevertheless there is a difference if only the consistency of the rules 1 and 2 is considered. They are logically consistent, but relative to rule 3 they are inconsistent. The following examples illustrate that it can even make a difference for the set of all rules, whether a rule is part of a set that is evaluated as to its logical consistency, or that this rule is taken as part of the constraints:

EXAMPLE 4A

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
rule_3 = *minor(x) ⇒ *~thief(x)
```

EXAMPLE 4B

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
c = {rule_3: *minor(x) ⇒ *~thief(x)}
```

We have seen in example 2 that the rules 1 and 2 by themselves are *logically* inconsistent. This logical inconsistency is maintained if rule 3 is added to the rules 1 and 2, because the resulting set still has an inconsistent subset.

However, the situation changes if rule 3 is taken *as the background* against which the consistency of the rules 1 and 2 is evaluated, as in example 4B. The conditions of the rules 1 and 2 are not compatible relative to a background consisting of rule 3. As a consequence the rules 1 and 2 are consistent *relative to this background*, even thought the rules 1 to 3 are logically inconsistent. Apparently it makes a difference whether a rule is considered as part of the set that is evaluated as to its logical consistency, or as part of the background for the consistency of the other rules.

In the examples 3A and 3B it did not matter for the consistency of the set whether rule 3 was part of the set, or part of the background, while in the examples 4A and 4B this was very relevant. This difference can be explained by pointing out that in the examples 4A and 4B, rule 3 made the *conditions* of the rules 1 and 2 incompatible, while in the examples 3A and 3B, rule 3 made the *conclusions* of the rules 1 and 2 incompatible.

This is an important point regarding the influence of the background on the consistency of a set of rules. The more demanding the background, the more strict are the constraints on the states of affairs that are compatible. If a set of states of affairs is incompatible relative to a certain background c, it will be incompatible relative to any background c' which imposes more constraints than c. [11]

The consistency of a set of rules varies positively with the compatibility of the conclusions of these rules, and negatively with the compatibility of the rule conditions. As a consequence, the addition of constraints to the background contributes to the consistency of rules by making the rule conditions incompatible. Addition of constraints detracts from the consistency of rules by making the rule conclusions incompatible. The overall effect of adding to the background of constraints on the consistency of a set of rules depends of the conditions and conclusions of the rules that are evaluated as to their consistency and the contents of the constraints that are added to the background.

5 Conditionless rules

Until now we have only considered conditional rules. There are also rules without conditions, such as the rule that it is forbidden to steal. Such rules share some characteristics with rules that have conditions, in particular that they can have exceptions. For the evaluation of their consistency they seem a little different, however. The first part of the definition of rule consistency, that there is a set of compatible states of affairs that satisfies the conditions of all the rules, seems not to apply to conditionless rules.

This seeming complication is easily remedied, however, by treating conditionless rules as rules with conditions, where the conditions are always satisfied. If conditions which are always satisfied are denoted by the term `*true`, conditionless rules are represented as rules with `*true` as their condition part. For instance:

[11] This observation does not hold without restrictions if exception-introducing constraints are added to the background.

```
*true ⇒ *od(x, ¬steal)
```

The rules that it is forbidden to steal (all actors ought to refrain from stealing) and that it is permitted to steal are then easily shown to be inconsistent against the background of the constraint that an action cannot both be forbidden and permitted[12]:

```
rule_1 = *true ⇒ *od(x, ¬steal)
rule_2 = *true ⇒ *pd(x, steal)
c = {*od(x, ¬action) ⇒ *~pd(x, action)}
```

The inconsistency of the rules 1 and 2 against the background c is illustrated by any case, since any case satisfies the conditions of these two rules.

6 Exceptions to rules

It is not uncommon that two rules in a legal system attach incompatible consequences to a case. For instance, the rule that an owner is allowed to do anything he likes with his property collides with many rules that limit his property right. In such cases, the law contains a *prima facie* rule conflict. Many prima facie rule conflicts turn out not to be *actual* conflicts, because one of the prima facie conflicting rules is left out of application by making an exception to it. Two or more rules are in actual conflict when they *actually* apply to one and the same case, and attach incompatible consequences to this case.

I will use an example again to sharpen our intuitions concerning the effect of exceptions to the consistency of rules. Take the following three rules:

1: Thieves are punishable.

2: Minors are not punishable.

3: In case of minors there is an exception to the rule that thieves are punishable.

These rules interact in case of a minor who is a thief. If rule 3 is left out of consideration, the rules 1 and 2 are inconsistent, because they lead to incompatible results in case of a minor thief. Rule 3 prevents that rule 1 is applied, however, so that the prima facie rule conflict is not actualised.[13] Therefore the rules 1 and 2 are in my view consistent against the background of rule 3, although they are logically inconsistent.

The insight that rules can have exceptions which prevents them to come into an actual conflict leads to the following adapted version of the above theory of rule consistency:

The rules in a set s are consistent relative to a set of constraints c if and only if it is not so that there are a subset s' of s and a possible case f such that

a. *the states of affairs in f are compatible relative to c,*

[12] Some would want to include this constraint into the set of logical constraints. In general the example leaves a lot to be said concerning deontic logic. This is beyond the scope of this paper, however.

[13] A logical account of the operation of exceptions to rules can be found in Hage 1997.

b. *the states of affairs in f satisfy the conditions of all the rules in s',*

c. *there is no exception to either one of the rules in s', and*

d. *the rules in s' attach consequences to f that are incompatible relative to c.*

Exceptions ought to be exceptional. I take this to mean that there are no exceptions to rules unless there are special reasons to make them. Such reasons exist if there are rules that attach the presence of an exception to a rule to the presence of some facts. For instance, if somebody is a minor, there is an exception for this person to the rule that thieves are punishable. Since this exception holds, in principle, for all minors, there is a rule[14] to the effect that in case of minors there is an exception to (amongst others) the rule that thieves are punishable.

In general I propose the theory that there can only be an exception to a rule if there is another rule, the conditions of which are satisfied, and which is not subject to an exception itself, that has as its conclusion that there is an exception to the first mentioned rule. This theory will be formalised in section 12. Exceptions that do not satisfy the above mentioned constraint are called 'free-floating exceptions'. A good theory about rule exceptions should, in my opinion, make such free-floating exceptions impossible.

The implications of the amendment to the theory of rule consistency which takes exceptions into account are illustrated by the following examples:

EXAMPLE 5

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
c = {rule_3 = *minor(x) ⇒ *exception(*thief(x) ⇒ *punishable(x))}
```

Rule 3 holds that if somebody is a minor, there is an exception for him to the rule that thieves are punishable.

The rules 1 and 2 by themselves are logically inconsistent. Inclusion of rule 3 in the background makes that if the conditions of rule 2 are satisfied, there is an exception to rule 1, which takes the rule conflict away. As a consequence, the rules 1 and 2 are consistent relative to a background that consists of rule 3.

[14] The conclusion that there is a rule is a somewhat strong rendering of the insight that judgments that there is an exception are universalizable.

Exceptions can also make a consistent set of rules inconsistent:

EXAMPLE 6

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
c = {rule_3 = *minor(x) ⇒ *exception(thief(x) ⇒ *punishable(x)),
     rule_4 = *second_offender(x) ⇒ *exception(rule_3)}
```

We have seen in example 5 that the rules 1 and 2 are consistent against the background consisting of rule 3. The addition of rule 4 to the background makes that there is no guarantee anymore that there is an exception to rule 1 in case of a minor. This is illustrated by the case that John is not only a thief and a minor, but also a second offender. In that case there is an exception to rule 3, and presumably no exception to rule 1. This illustrates that if rule 4 is added to the background, there are possible cases in which the conditions of both the rules 1 and 2 are satisfied, and in which these two rules are in actual conflict.

7 Deontic consistency

Rule consistency is not the same thing as deontic consistency, but yet some of the results that were obtained above can be put to use in connection with deontic consistency. This is not the place to give the subject of deontic consistency an extensive treatment. The interested reader is referred to the literature mentioned in footnote 1. Instead I will just offer some brief remarks on the subject, to introduce the main point.

Deontic sentences can be subdivided into sentences of the ought-to-be type and sentences of the ought-to-do type. States of affairs of the ought-to-be type are incompatible relative to a set of constraints c, if either some state of affairs both is permitted and forbidden to be the case, or some states of affairs that are incompatible relative to c, ought to be the case. For instance, the states of affairs that the car ought to be in the parking lot and that it is permitted not to be in the parking lot are incompatible, and also the states of affairs that a car both ought to be inside and outside the parking lot, given the constraint that a car cannot both be inside and outside a parking lot.

Sentences of the ought-to-do type differ fundamentally from sentences of the ought-to-be type because they refer to sets of actors. They say about actors in the actorset that they are obligated/permitted/forbidden to perform or refrain from performing some type of action. For instance they may say that car drivers ought to turn on the car lights.

With regard to the issue of compatibility the actor sets in ought-to-do sentences fulfill a role that is similar to that of the conditions of rules. Intuitively two ought-to-do states of affairs are incompatible if it is possible that an actor belongs to the actor sets of both states of affairs, while it is impossble that this actor fulfills the demands imposed on him by both deontic states of affairs. For instance the states of affairs that all Belgian people ought to carry a legitimation and that all Dutch people are forbidden to carry a legitimation are incompatible since it is possible that somebody has both the Dutch and the Belgian nationality, while it is impossible both to carry and not to carry a legitimation.

With this example the first, informal, part of this paper is finished. In the second part of the paper, the introduced notions will be formalised and made more precise.

PART 2

8 The language L_{RL}

The notion of rule consistency will be formalised by model-theoretic means, that is in terms of possible worlds. Intuitively, a set of rules is inconsistent if there is a possible world in which the conditions of all the rules are satisfied, and there is no exception to either one of the rules, while there is no possible world in which the conclusions of all the rules obtain. Variations on the notion of consistency are realised by different characterisations of possible worlds.

In order to give such a formalisation I need to be able to talk about states of affairs and rules in the logical language that I employ. For that purpose I will use a special logical language L_{RL}, the language of *Rule Logic*. L_{RL}, is essentially the language of first order predicate logic, augmented with some conventions.[15] I assume that all formulas of L_{RL} have an uppercase initial, while all terms (including function expressions) have a lowercase initial. Variables of L_{RL} are *italicised*. For the meta-language of L_{RL} I use schematic sentences and terms, unless the contrary is clear from the context.

8.1States of affairs

I adopt the convention that all well-formed formulas of L_{RL} *express* states of affairs. For every state of affairs expressible in L_{RL} there is a term that *denotes* it. In the case of atomic formulas this term is constructed by replacing the uppercase initial of the formula expressing the state of affairs by the corresponding lowercase initial and prefixing the resulting string by an asterisk (*). E.g. if `Thief(john)` expresses the state of affairs that John is a thief, `*thief(john)` denotes the same state of affairs.[16] All terms that start with an asterisk are assumed to denote states of affairs. In this way the terms of L_{RL} are subdivided into terms that denote states of affairs and terms that denote other individuals.

[15] L_{RL} is a subset of the language of Reason-based Logic. This language is described and motivated in Hage 1997, ch. IV.

[16] Notice that this term does *not* denote the sentence 'Thief(john)'. The state of affairs expressed by a formula is not the reification of that formula. This means that formulas referring to states of affairs need not belong to a meta-language.

In the case of logically compound sentences, the replacement of uppercase letters by lowercase ones is executed for all atoms that are part of it. The term `*it's_raining ∨ shines(sun)` denotes the disjunctive state of affairs that either it's raining or the sun shines.[17]

States of affairs denoted by terms that do not contain free variables are called *concrete* states of affairs. States of affairs denoted by terms that contain one or more free variables are *generic* states of affairs.

8.2Constraints and rules

Some kinds of states of affairs tend to go together, while other ones exclude each other. For instance, the states of affairs that *x* kisses *y* tends to go together with the state of affairs that *x* touches *y*[18], and the state of affairs that *x* is a circle tends (very strongly) to exclude the state of affairs that *x* is a square. These relations between (usually generic) states of affairs are called *constraints* on states of affairs.

Rules, including legal rules, are a special kind of constraints. The rule that thieves are punishable makes that the state of affairs that somebody is a thief goes together with the state of affairs that this person is punishable. The unconditional rule that it is forbidden to steal makes that every state of affairs goes together with the state of affairs that it is forbidden to steal. The power-conferring rule that the government and the parliament together are competent to make laws, makes that the states of affairs that these bodies are the government and parliament go together with the state of affairs that these bodies are competent to make laws.

The symbol ⇒ is used to denote constraints in general and rules in particular. ⇒ is an infix functor with two parameters, both for (usually generic) states of affairs. The first parameter stands for the rule conditions, the second parameter stands for the rule conclusion. If a rule has no conditions, the condition part is taken by the term `*true`. Examples of well-formed rule expressions are:

`*minor(`*x*`) ⇒ *~punishable(`*x*`)`, and

`*true ⇒ *fd(x, steal)`

Rule formulations are according to these conventions not well-formed sentences of Rule Logic. From a logical point of view they are function expressions that denote individuals.

It is possible to make statements about rules, such as the statement that a rule exists. The predicate `Exists` serves to express that a rule exists. It is defined by the following sentence:

`Exists(`*rule*`) ≡def. ∃`*x*`(`*x* `=` *rule*`)`

17 This means that the symbols for the logical operators are also used for functors that map states of affairs on more complex states of affairs, such that the relation between logically compound sentences and the corresponding compound states of affairs is maintained.

18 This example stems from Barwise and Perry 1983 p. 12.

Finally, the language L_{RL} has a one-place predicate `Exception` that ranges over instantiated rules and expresses that there is an exception to the rule in question for the case to which the rule is instantiated. For instance:

`Exception(*thief(john)` $\Rightarrow$ `*punishable(john))`

9 Model theory for Rule Logic

Model-theoretic semantics for logic specifies the meanings of logical operators by means of the truth conditions of sentences in which these operators occur. Since rules are assumed[19] to have no truth values, and therefore also no truth conditions, this kind of semantics does not work for rules. Nevertheless it is desirable to have a formalism that can both account for sentences that are *about* rules, such as that a rule exist, or that there is an exception to a rule, and for the effects of rule application. Because the formalism that I develop for this purpose is strongly inspired by the usual model-theoretic semantics for predicate logic, I call it a model theory for rules.

Model-theoretic semantics traditionally focuses on the truth values of sentences. I will present the model theory in such a way that the emphasis is on the states of affairs that obtain. For instance, a world only counts as logically possible if it satisfies the constraint that if a state of affairs of the form `*a & b` obtains, the states of affairs of the forms `*a` and `*b` must also obtain, and vice versa. There is a close connection between this approach and traditional model-theoretic semantics, because a sentence that expresses a state of affairs is true if and only if this state of affairs obtains. I prefer the emphasis on states of affairs, because the effects of rules are in the first place that states of affairs obtain, and only in the second place that particular sentences are true.

A second difference, in line with the first, is that I do not take the notion of a possible world for granted. Traditionally model-theoretic semantics specifies which relations exist between the truth values of sentences in a possible world. I turn this around and specify which relations between states of affairs must hold for a world to be a possible one. In this way the function of rules as constraints on possible worlds is highlighted. It makes it also easier to distinguish different notions of possibility as defined by different sets of constraints that are taken into account. We have seen in section 3 how variants of rule consistency depend on various notions of possibility and compatibility. These distinctions can be treated more naturally in a theory that focuses on possibility rather than on relations between truth values.

Central in the model theory for rules is the notion of a constraint. Logical constraints hold in general for all *logically possible worlds*. These are, in the present context, the constraints of predicate logic, augmented with one additional constraint that characterises the logic of rules. Together, these constraints on all logically possible worlds are called the constraints of *Rule Logic*. I will treat rules as

[19] In the present paper I will not argue for this assumption, except for pointing out that the lack of a truth value immediately follows from the treatment of rules as logical individuals.

local constraints. Rules define what is possible and impossible *in a concrete world*, that is, in the worlds in which these rules exist.

I will present the model-theoretic characterisation of rule logic in two stages. First I disregard that exceptions should be minimised. This leads to a relatively simple characterisation that has the drawback that exceptions to rules are possible, even if there are no reasons for their presence. Then, in section 13, I will formulate an additional constraint that takes the minimisation of exceptions into account. This modification makes use of methods familiar from nonmonotonic logic, and introduces additional technicalities.

The first model-theoretic characterisation of worlds that are possible according to Rule Logic runs as follows:

CONSTRAINTS ON WORLDS THAT ARE LOGICALLY POSSIBLE ACCORDING TO L_{RL}

Let L_{RL} be the language of Rule Logic. $L_{RL} = \{S_1, S_2, ..., S_n\}$, where S_1 ... S_n are all the well-formed sentences of L_{RL}.

Let S_i be a sentence in L_{RL}, and let $*sa_i$ denote the state of affairs that is expressed by S_i. $*sa_i$ is then a state of affairs that is *possible relative to* L_{RL}.[20]

Let the set SA be the set of all states of affairs that are possible relative to L_{RL}, and let W be the power set of SA. Intuitively, W stands for the set of all worlds, the content of which is expressible in L_{RL}. Every $w \in W$ is a subset of SA.

There are no other constraints on the states of affairs that are elements of the worlds in W. There are, for instance, worlds in W in which the state of affairs $*p \ \& \ q$ obtains, but in which the state of affairs $*q$ does not obtain. Such worlds are possible relative to L_{RL}, but they are not logically possible according to Rule Logic. Worlds that are logically possible are subject to a number of additional constraints. The set of these logically possible worlds is denoted by W_{RL}.

CONSTRAINTS ON WORLDS THAT ARE LOGICALLY POSSIBLE ACCORDING TO RULE LOGIC

1. if $*p \in w$ then $*{\sim}p \notin w$, and if $*{\sim}p \in w$, then $*p \notin w$.[21]
2. $*p \ \& \ q \in w$ if and only if both $*p \in w$ and $*q \in w$.
3. $*p \vee q \in w$ if and only if either $*p \in w$, or $*q \in w$, or both.
4. $*p \rightarrow q \in w$ if and only if either $*p \notin w$, or $*q \in w$, or both.
5. $*p \equiv q \in w$ if and only if either both $*p \in w$ and $*q \in w$, or both $*p \notin w$ and $*q \notin w$.

---

[20] L_{RL} may be thought of as the conceptual scheme by means of which worlds are 'captured'.

[21] Notice that it is not necessary that a possible world either contains $*p$ or $*{\sim}p$.

These constraints correspond to the traditional constraints of propositional logic stated in terms of relations between states of affairs.

6. $*\exists x(r(x)) \in$ w if and only if there is an individual a, such that $*r(a) \in$ w.
7. $*\forall x(r(x)) \in$ w if and only if there is no individual a, such that $*r(a) \notin$ w.

These constraints give the traditional meaning of the quantifiers, again stated in terms of states of affairs.[22] The worlds that satisfy these constraints and constraint 9 to be mentioned in a while, are called the possible worlds of predicate logic, to be abbreviated as W_{PL}.

A constraint that is characteristic for Rule Logic is that if the conditions of an existing rule are satisfied and there is no exception to this rule, the consequences of this rule obtain. Let *conditions/σ and *conclusion/σ denote the states of affairs expressed by respectively the conditions and the conclusion of a rule with their variables instantiated according to substitution σ. Then the above mentioned constraint becomes:

8. If *exists(*conditions ⇒ *conclusion) ∈ w, and
   *conditions/σ ∈ w, and
   *exception(*conditions/σ ⇒ conclusion/σ) ∉ w, then
   *conclusion/σ ∈ w.

Finally there is a constraint to guarantee that terms that denote states of affairs expressed by logically equivalent sentences are co-referential:

9. If and only if for all worlds $w \in W_{RL}$ it holds that $*p \equiv q \in w$, then $*p = *q$.

10 Compatibility of states of affairs

Given the model theory for Rule Logic, it is possible to give a formal characterisation of rule consistency. The starting point is the characterisation of compatible states of affairs:

RELATIVE COMPATIBILITY OF STATES OF AFFAIRS

Let c be a set of constraints, and let W_c be the set of worlds $w \in W_{RL}$, such that for every constraint $c_i \in c$, it holds that $*exists(c_i) \in w$. The states of affairs in a set s are then said to be compatible relative to the set of constraints c if and only if there is some set of states of affairs $s' \in W_c$ such that $s \subseteq s'$.

[22] To gain simplicity at the cost of precision, the formulations of the constraints 6 and 7 do not deal with compound formulas, or the use of quantifiers or function expressions within the scope of the quantifiers.

LOGICAL COMPATIBILITY OF STATES OF AFFAIRS

The states of affairs in a set s are logically compatible if and only if they are compatible relative to the empty set of constraints: $s \in W_{\varnothing}$.

Let me illustrate this by means of the following examples[23]:

<u>Example 7</u>

`s = {*punishable(john), *~punishable(john)}`

This set is logically incompatible, because of the first constraint on logically possible worlds.

<u>Example 8A</u>

`s = {*thief(john), *~punishable(john)}`

This set is obviously logically compatible, because there is no constraint on logically possible worlds that prevents the co-occurrence of these states of affairs.

We have seen that it is also possible to define a notion of compatibility that treats constraints, including rules, as a kind of background relative to which compatibilities are judged. If such a background contains the rule that thieves are punishable, the states of affairs that somebody is a thief and that he is not punishable, are incompatible relative to this background. The compatibility can then be restored by adding the presence of an exception to the rule that thieves are punishable to the set of states of affairs that is evaluated as to its compatibility, or to the background of constraints. This is illustrated by the following two examples:

<u>Example 8B</u>

s = {`*thief(john), *~punishable(john)`}

c = {`*thief(`*x*`)` ⇒ `*punishable(`*x*`)`}

The states of affairs in s are incompatible relative to c.

To be sure, the states of affairs in s are logically compatible, but the conditions of the rule that thieves are punishable which is an element of c, are satisfied in all worlds in which the states of affairs of s obtain. This rule leads to the conclusion that John is punishable.

If the set s is to be compatible relative to c, s should contain an exception to the rule that thieves are punishable. The following set s is compatible relative to c, because c contains the necessary, although strange, exception:

[23] The examples should really deal with the closure of sets of states of affairs in the sense of section 11, rather than with the sets themselves. To keep this section relatively simple, I ignore this observation.

Example 8C

```
s = {*thief(john), *~punishable(john), *minor(john)}
c = {*thief(x) ⇒ *punishable(x),
     *minor(john) ⇒ *exception(thief(john)⇒ punishable(john))}
```

11 The consistency of rules and deontic sentences

By means of the notions of logical compatibility of states of affairs and compatibility of states of affairs relative to a set of constraints, it is possible to give a formal characterisation of rule consistency.

Let r = {r_1 ... r_n} be a finite set of n rules, where r_i = *$conditions_i$ ⇒ *$conclusion_i$, for i = 1 to n.

Let s = {σ_1 ... σ_n} be a set of n instantiations for the variables that occur in r, where σ_i is applied to the variables in r_i. For instance, let r_3 be `*thief(x) ⇒ *punishable(x)`, and let σ_3 be {`x -> john`}. Then the instantiation of r3 by means of σ3, inst(r_3, σ_3), is

`*thief(john) ⇒ *punishable(john)`.

Let $I_{conditions}$(r, σ) be the set of the instantiations by means of σ of the conditions of all rules in r. That is:

$I_{conditions}$(r, σ) = {inst($conditions_1$, σ_1), ... inst($conditions_n$, σ_n)}.

Let $I_{conclusion}$(r, σ) be {inst($conclusion_1$, σ_1), ... inst($conclusion_n$, σ_n)}.

Let $I_{\sim exception}$(r, σ) be

{*~exception(inst($conditions_1$, σ_1) ⇒ inst($conclusion_1$, σ_1)) ...

*~exception(inst($conditions_n$, σ_n) ⇒ inst($conclusion_n$, σ_n))}.

Then the following definition captures the notion of rule consistency relative to a set of constraints:

RELATIVISED RULE CONSISTENCY:

The rules in the set s are consistent relative to a set of constraints c, if and only if it is not so that there is a set s' ⊆ s and a set of instantiations σ, such that

a. the set $I_{conditions}$(s', σ) ∪ $I_{\sim exception}$(r, σ) is compatible relative to c ∪ s,

b. the set $I_{conclusion}$(r, σ) is incompatible relative to c ∪ s.

The compatibility of the joint rule conditions and conclusions, and the absence of exceptions to the rules is judged against the background of both the set of constraints and the rules themselves, because the rules that are evaluated as to their consistency also impose constraints on the world in which they exist.

LOGICAL CONSISTENCY OF RULES

The rules in the set s are logically consistent, if and only if it is not so that there is a set s' $\subseteq$ s and a set of instantiations σ, such that

a. the set $I_{conditions}(s', \sigma) \cup I_{exception}(r, \sigma)$ is compatible relative to s,

b. the set $I_{conclusion}(r, \sigma)$ is incompatible relative to s.

Let me re-use some examples of the sections 3 and 4 to illustrate these definitions:

EXAMPLE 1

`rule_1 = *thief(x) ⇒ *punishable(x)`

`rule_2 = *thief(x) ⇒ *~punishable(x)`

The rules 1 and 2 are logically inconsistent, as is illustrated by the set of instantiations {σ1, σ2}, where σ1 = σ2 = {$x \rightarrow$ `john`}.

EXAMPLE 3A

`rule_1 = *thief(x) ⇒ *punishable(x)`

`rule_2 = *minor(x) ⇒ *protected(x)`

`rule_3 = *protected(x) ⇒ *~punishable(x)`

That the rules 1, 2 and 3 are logically inconsistent is illustrated by the set of instantiations {σ1, σ2, σ3}, where σ1 = σ2 = σ3 = {$x \rightarrow$ `john`}, with c = ∅.

EXAMPLE 3B

`rule_1 = *thief(x) ⇒ *punishable(x)`

`rule_2 = *minor(x) ⇒ *protected(x)`

`c = {*protected(x) ⇒ *~punishable(x)}`

The rules 1 and 2 are also inconsistent relative to c because there can be no instantiation of x that makes the instantiations of the states of affairs `*punishable(x)` and `*protected(x)` co-obtain in a world in which the constraint in c exist.

EXAMPLE 4A

`rule_1 = *thief(x) ⇒ *punishable(x)`

`rule_2 = *minor(x) ⇒ *~punishable(x)`

`rule_3 = *minor(x) ⇒ *~thief(x)}`

The rules 1 to 3 are logically inconsistent, because it is logically possible that somebody is both a thief and a minor, while it is logically impossible that somebody both is and is not punishable. Notice that rule 3 has no influence on the consistency of the set as a whole. The inconsistency is caused by the rules 1 and 2, and cannot be removed by rule 3.

EXAMPLE 4B

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
c = {*minor(x) ⇒ *~thief(x)}
```

We have seen that the three rules taken together are *logically* inconsistent. However, the conditions of the rules 1 and 2 are not compatible relative to c, because there can be no instantiation of `x` that makes the states of affairs `*thief(x)` and `*minor(x)` co-obtain in a world in which the constraints in c exist. Therefore the rules 1 and 2 are consistent against the background of c.

EXAMPLE 5B

```
rule_1 = *thief(x) ⇒ *punishable(x)
rule_2 = *minor(x) ⇒ *~punishable(x)
c = {*minor(x) ⇒ *exception(thief(x) ⇒ punishable(x))}
```

The rules 1 and 2 are consistent relative to c, because there can be no instantiation of *x* that makes the state of affairs `*minor(x)` obtain in a world in which the constraint in c holds, and in which the state of affairs `*exception(thief(x) ⇒ punishable(x))` does not obtain. As a consequence, there can be an instantiation of x such that `*thief(x)` and `*minor(x)` co-obtain in a world in which the rules 1 and 2 both exist in combination with the constraint in c.

12 The consistency of deontic sentences

I will use the predicate Ob, which stands for ought-to-be, to operate on states of affairs. The sentence `Ob(*a)` then means that the state of affairs `*a` ought to be the case. (E.g. the car ought to be in the parking lot). Similarly, `Pb` stands for permitted-to-be, and `Fb` stands for forbidden-to-be.

Given these definitions, the following pairs of schematic states of affairs are logically incompatible[24]:

```
*ob(a) - *pb(~a)
*ob(~a) - *pb(a)
*fb(a) - *pb(a)
*fb(~a) - *pb(~a)
```

[24] An amendment to this effect should be made to the constraints of Rule Logic.

I will use the predicate Od, which stands for ought-to-be, to operate on pairs of a set of actors, and an action type. Similarly, Fd stands for forbidden-to-do, and Pd stands for permitted to do. For example, `Pd({john}, steal)` would mean that John is permitted to steal.

The operator ¬ works on terms that denote action types and is meant to map action types on the types consisting of refraining from this action. So, if `steal` denotes the action type of stealing, `¬steal` denotes the action type refraining from stealing. For example, `Od({john}, ¬steal)` would mean that John ought to refrain from stealing.

The pairs of schematic states of affairs

*od(*actorset*, *actiontype*) - *pd(*actorset*, ¬*actiontype*)

*od(*actorset*, ¬*actiontype*) - *pd(*actorset*, *actiontype*)

*fd(*actorset*, *actiontype*) - *pd(*actorset*, *actiontype*)

*fd(*actorset*, ¬*actiontype*) - *pd(*actorset*, ¬*actiontype*)

*fd(*actorset*, *actiontype*) - *od(*actorset*, *actiontype*)

*fd(*actorset*, ¬*actiontype*) - *od(*actorset*, ¬*actiontype*)

are assumed to be *logically* incompatible.[25]

I will use the two-place predicate Performs to indicate that an actor performs some type of action. So `Performs(john, steal)` means that John steals.

To facilitate the formulation a theory of compatibility of deontic states of affairs of the ought-to-do type, I propose to reduce sentences of the forbidden-to-do and the permitted-to-do type to sentences of the ought-to-do type as follows:

```
Pd(actorset, action) ≡ ~Od(actorset, ¬action)
Fd(actorset, action) ≡ Od(actorset, ¬action)
```

Given these conventions, the compatibility of deontic states of affairs relative to a set of constraints can be expressed as follows:

The states of affairs

`*od(actorset1, actiontype1)` and
`*od(actorset2, actiontype2)`

are incompatible relative to a set of constraints c if and only if the states of affairs

[25] An amendment to this effect should be made to the constraints of Rule Logic.

`*actor ∈ actorset1` and
`*actor ∈ actorset2`

are compatible relative to c, while the states of affairs

`*performs(actor, actiontype1)` and
`*performs(actor, actiontype2)`

are incompatible relative to c.

For example, the states of affairs

`*od(car_drivers, turn_on_car_lights)` and
`*od(smugglers, leave_out_car_lights)`

are incompapble relative to the constraint that the action types of turning the car lights on and leaving them out are are incompatible.[26] The reason for this is that somebody may be both a car driver and a smuggler, and for these persons, the two deontic states of affairs demand incompatible actions.

13 Minimising exceptions

Arguably there are no exceptions to rules whose conditions are satisfied, unless there is a special reason for it. Such a reason consists of facts that are made into an exception by some other rule. It is possible to modify the constraints on worlds that are possible according to Rule Logic to take this into account. The modification must make sure that there are no 'free floating' exceptions to rules, exceptions that are not based on exception creating rules and additional facts that satisfy their conditions.

The result of such a modification is that the number of exceptions is minimised to those that are necessary because of the other facts and rules that obtain in the world, that is to the so-called 'grounded' exceptions. Minimisation of exceptions is a logical technique that is widely employed in the study of so-called nonmonotonic logics. The following modification of the constraints on Rule Logic possible worlds was inspired by the research on nonmonotonic logic, especially by the way in which fixed point techniques are used for the definition of extensions of a theory (see Reiter 1980).

The modification is that the following constraint, which will be explained immediately after its formulation, is to be added to the constraints on worlds in W_{RL}.

10. For every world $w \in W_{RL}$ there must be a grounded sequence $w_0 \ldots w$, all elements of which are in W_{PL}, such that there is no state of affairs $s \in w_0$ of the form `*exception(conditions/σ ⇒ conclusion/σ)`.

The explanation of this constraint comes in two steps. The first step is that a subset of W, called W_{PL}, is defined,. This subset consists of all worlds in W that satisfy the constraints 1-7 and 9 of Rule Logic.

26 To avoid additional technicalities, I refrain here from formalising the compatibility of action types.

This set W_{PL} is my starting point for the new characterisation of the set of possible worlds W_{RL}, which is the second step. To specify this last set, I need some auxiliary constructs.

The first construct is a function that operates on possible worlds and maps them on the worlds that are the possible world-counterpart of their deductive closure. I will call the function CL (closure), and it is defined as follows:

Let $w \in W$, and s stand for a state of affairs. Then it holds that

$CL(w) = \{s \mid s \in \cap w_i \text{ for all } w_i \in W_{PL}, \text{ such that } w \subseteq w_i\}$

Informally: The closure of a possible world w is the intersection of all PL-possible worlds of which w is a subset.[27]

The second construct is a three-place relation R over worlds in W_{PL}, which is defined as follows:

$R(w_1, w_2, w_3)$ is true, if and only if $w_2 = CL(w_1 \cup \text{additionset})$, where

additionset =

{ `*conclusion/`σ |

`*exists(conditions` ⇒ `conclusion)` $\in w_1$,

`*conditions/`σ $\in w_1$, and

`*exception(*conditions/`σ ⇒ `*conclusion/`σ`)` $\notin w_3$ }

Intuitively this boils down to the following: For every rule that exists in world w_1, the conditions of which are satisfied in w_1, and to which there is no exception in w_3, the state of affairs that this rule's conclusion holds is added to the states of affairs of w_1. The result leads to w_2, which is the closure of the result of the application of all applicable rules in w_1.

The third auxiliary construct is that of finite sequences S_i of worlds. Such a sequence consists of one or more worlds in W_{PL}, denoted as $S_i[1] \ldots S_i[n]$, with $n \geq 1$. The worlds in such a sequence satisfy the constraint that $R(S_i[n], S_i[n+1], w)$, where w is some element of W_{PL}. This world w is called the *reference world* of the sequence. The world $S_i[1]$ is the *starting world* of sequence S_i.

Given the definition of the relation R, it holds for every i and every n such that $S_i[n+1]$ exists, that $S_i[n] \subseteq S_i[n+1]$. Because all elements of a sequence must be in W_{PL}, not every world in W_{PL} can be the starting world of a sequence in combination with every reference world. For instance, the world {`*a`, `*b`, `*exists(*a` ⇒ `*~b)`} cannot be a starting world in combination with a reference world that does not contain the state of affairs `*exception(*a` ⇒ `*~b)`.

Intuitively a sequence represents a series of possible worlds in which each world is the result of applying all applicable rules in its predecessor. The interesting case for my purposes is when a sequence ends with a subsequence of at least two identical worlds. When this happens, the final worlds in the

[27] Notice that if $w \notin W_{PL}$, then $Cl(w) = \emptyset$.

sequence are worlds in which all applicable rules have been applied, and the results do not give rise no new possibilities for rule application. Let us call such sequences *halting sequences*. The final world in a halting sequence is called the *fixed point* of that sequence. For a fixed point w_f it holds that $R(w_f, w_f, w)$.

A subset of the halting sequences consists of those sequences in which the fixed points are identical to their reference world. I will call these sequences *grounded sequences*. Those fixed points are then worlds in which no new rules can be applied, given the exceptions to rules that obtain in these worlds themselves. For these fixed points w_f it obviously holds that $R(w_f, w_f, w_f)$.

Since worlds in W_{RL} should not contain exceptions that do not ultimately result from the application of existing rules to facts, an additional requirement must be formulated: every world in W_{RL} must be the halting world of a grounded sequence, which starts with a world that does not contain any exceptions. The existence of the grounded sequence then shows that repeated application of the rules in the 'starting world' of the sequence results in the world that is to be in W_{RL}. In other words, the last world can be the result of repeatedly applying the rules in a world that does not contain exceptions.

These requirements on worlds in W_{RL} lead to the formulation of constraint 10, which guarantees that a world in W_{RL} does not contain free floating exceptions.

14 Summary

In this paper I have developed the theory that a set of rules is consistent if it is not possible that 1) the conditions of the rules in the set are all satisfied, 2) there is no exception to either one of the rules, and 3) the consequences of the rules are incompatible. To this purpose I generalised the notion of consistency to make it cover rules and relativised it to some background of constraints.

I introduced the constraints of Rule Logic as a minimal set of constraints that hold in all worlds that are logically possible. Rules are additional constraints that hold in the worlds in which they exist. It turned out to be possible to characterise formally the consistency of states of affairs and of rules on the basis of worlds that are possible according to Rule Logic.

The insights developed in connection with rule consistency turned out to be applicable to the compatibility of deontic states of affairs of the ought-to-do type too.

The formal theory could still be refined by not allowing free floating exceptions to rules. To that purpose an additional constraint was added on worlds that are possible according to Rule Logic.

References

Alchourrón, C.E. (1993). Philosophical Foundations of Deontic Logic and the Logic of Defeasible Conditionals, in J-J. Ch. Meijer and R.J. Wieringa (eds.), *Deontic Logic in Computer Science*. John Wiley & Sons, Chicester, pp. 43-84.

Barwise, J. and J. Perry (1986), *Situations and Attitudes*, MIT Press, Cambridge e.a.

Chellas, B.F. (1980). *Modal logic; an introduction*. Cambridge University Press, Cambridge e.a.

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

Haan, N. den (1996). *Automated Legal Reasoning*. Doctoral Thesis, Amsterdam.

Hage, J.C. (1997). *Reasoning with rules*, Kluwer Academic Publishers, Dordrecht.

Hage, J.C. (1999). Rule Consistency. In H.J. van den Herik e.a. (eds.), *Legal Knowledge Based Systems. Jurix 1999: The Twelfth Conference*, GNI, Nijmegen, p. 47-60.

Hage, J.C. (forthcoming). Rule Consistency. *Law and Philosophy* (in press).

Hamner Hill, H. (1987). A functional taxonomy of normative conflict. *Law and Philosophy* 6, p. 227-247.

Kelsen, H. (1979). *Allgemeine Theorie der Normen*. (K. Ringhofer und R. Walter hrsg.) Manz-Verlag, Wien.

Lindahl, L. (1992). Conflicts in Systems of Legal Norms: A Logical Point of View. P.W. Brouwer et al. (eds.), *Coherence and Conflict in Law*, Kluwer Deventer, p. 39-64.

Prakken, H. and G. Sartor (1996). A Dialectical Model of Assessing Conflicting Arguments in Legal Reasoning. *Artificial Intelligence and Law*, vol. 4 no. 3/4. pp. 331-368. Also in H. Prakken and G. Sartor (eds.), *Logical Models of Legal Argumentation*. Kluwer Academic Publishers, Dordrecht 1997.

Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence* 13, pp. 81-132.

Ruiter, D.W.P. (1997). Legal validity qua specific mode of existence. *Law and Philosophy* 16, p. 479-505.

Strawson, P. F. (1971). *Logico Linguistic Papers*, Methuen, London.

Wright, G.H. Von (1991). Is There a Logic of Norms? *Ratio Juris*, vol. 4 no. 3, pp. 265-283.