the law are, at least partly, determined by new decisions. These decisions need not be coherent in content, as Raz correctly pointed out.[60]

A system that is coherent in origin need not be coherent in content and vice versa. If this is what Raz wanted to point out with his argument, he was right. It seems, however, that he meant to say more in two respects. He apparently adduced the point of several sources of law as an argument against coherentism in general. That would be a mistake, because the coherence of law might in Kelsenian vein be constructed as coherence of origin instead of content. Moreover, he seems to assume that the fact that the contents of legal decisions need not be coherent implies that the content of the law is not coherent. That would only follow on a view according to which the contents of the law are by and large determined by the contents of these decisions. Such a view is certainly defendable, but depends on a particular view how the law is to be constructed. An alternative view would be that the contents of these decisions should be considered as input to a process of (re)construction that leads to a system that is coherent in content.[61]

Let us take a step back from the discussion about the proper role of authoritative legal decisions in the construction of the law. The position one takes in this discussion will be influenced by one's views about the relation between the law and state authorities and between law and politics.[62] Integrated coherentism requires that ones politico-philosophical views in this respect cohere with one's views about the room for legal decision makers to make the law coherent in content. It is compatible with different views about the proper kind of coherence (origin or content) in constructing the law. It is even compatible with rejection of both content and origin coherentism concerning the law.

The discussion whether the law should be constructed coherently and if so on the basis of content or origin, deals with another form of coherentism than the broad version defended in this chapter. It puts stronger demands on the way law should be constructed than merely that the theory of the law is part of a coherent acceptance set as defined in section 5. In fact it is a discussion whether the law should be coherent in the strict sense and what

[60] It may intuitively be attractive to say that a system that is based on one single starting point, no matter whether it concerns content or origin, is for that reason coherent. However, the presence of a single starting point has more to do with simplicity than with coherence. I do not see why a system with several starting points that are suitably delimited in their sphere of operation should be less coherent. In this chapter, I will not develop this issue any further, however.

[61] Cf. Peczenik and Hage 2000.

[62] The discussion about coherentism between Dworkin and Raz clearly illustrates this point.

such strict coherence should look like (criteria for strict coherence). All answers in this discussion, including non-coherentist ones, are compatible with integrated coherentism, at least if one makes one's position in this discussion broadly coherent with one's other views, including especially those concerning the relation between the law and politics.

Raz's argument against constitutive coherentism seems to presuppose that the discussion about legal coherence can be treated independently of the rest of one's acceptance set. By treating the discussion as a local one, it seems that the argument based on different sources of authority can be adduced as an argument against coherentism in general. By taking a step back from this discussion and seeing it in the context of a broader discussion (a step toward global coherence), one can see why Raz's argument does not affect broad coherentism, but only one specific and local variant of strict coherentism, namely *content* coherence of the *law*.

10. CONCLUSION

The foundation of this chapter is a theory of justification. According to this theory a justification is an argument why something should be accepted rather than rejected, given what else is accepted. Coherentism as a theory about justification is, given the above definition necessarily internalist. Justification is justification of acceptances on the basis of acceptances and perceptive states. There are two variants of internalist theories of justification, namely foundationalist and coherentist, which seem to be mutually exclusive and jointly exhaustive. On closer inspection, however, it turns out that foundationalism can be incorporated in a broad form of coherentism and that broad coherentism is the single convincing theory of justification as it was defined. This broad form of coherentism merely holds that acceptances are to be justified by means of other acceptances and that none of them is a priori justified.

If one takes a closer look at the way coherentism functions in the practice of justification, it turns out that there is a natural tendency of acceptance sets to become more and more comprehensive, but also that a completely coherent set is an unattainable ideal. The pursuit of coherence functions in practice as a correctional device by means of which incoherent acceptance sets can be improved. The unattainable ideal is a completely coherent set, a set that contains all acceptances that it should contain according to the standards contained in the set itself and does not contain anything that according to these standards should be rejected.

A broadly coherent acceptance set may, but need not, demand that some part of it, for instance the part dealing with the law, is coherent in a more strict way. If it does, this demand should be broadly coherent with the rest of the complete acceptance set. Moreover, the standards for this more strict form of coherence should also cohere in the broad sense with the rest of the acceptance set as a whole. Theories as those of Dworkin (law as integrity) and Raz (merely a limited role for local coherence) can be seen as competing precisely on the issue whether and how the law should be constructed as coherent in a more strict sense. An argument that it should not be constructed as strictly coherent, should not be interpreted as an argument against broad coherentism, however. It is rather the case that such an argument should fit in a broadly coherent theory of everything.

Chapter 3

REASON-BASED LOGIC

1. REASON-BASED LOGIC AS AN EXTENSION OF PREDICATE LOGIC

‘Traditional’ logics such as propositional and predicate logic sketch a one-sided picture of what goes on in real life reasoning. Arguments of the form ‘modus ponens’ have a dominant place in this picture. Other forms of reasoning, such as arguments based on balancing reasons, can only with some ingenuity be modeled in these logics. Since such arguments play an important role both in the law and in practical reasoning in general, it is attractive to have a logic at one’s disposal that can deal with arguments based on the balancing of reasons for and against a conclusion.[1] *Reason-based Logic* (RBL) is such a logic.

One way to look at RBL is as a logic that is dedicated to practical (legal and moral) reasoning, with special attention to entities that function prominently in these types of reasoning, such as rules and principles. This is the way in which RBL was introduced in my *Reasoning with Rules*.[2] Another way to look at it, which I want to emphasize here, is as an extension of predicate logic. First order predicate logic is included in RBL and RBL adds to predicate logic a number of linguistic elements and axioms that deal with reasons. The ‘philosophy’ behind this second way of looking at RBL is

[1] Cf. Alexy 2003.

[2] The version of Reason-based Logic described in Hage 1996 can for the present purposes be equated with the version of *Reasoning with Rules*.

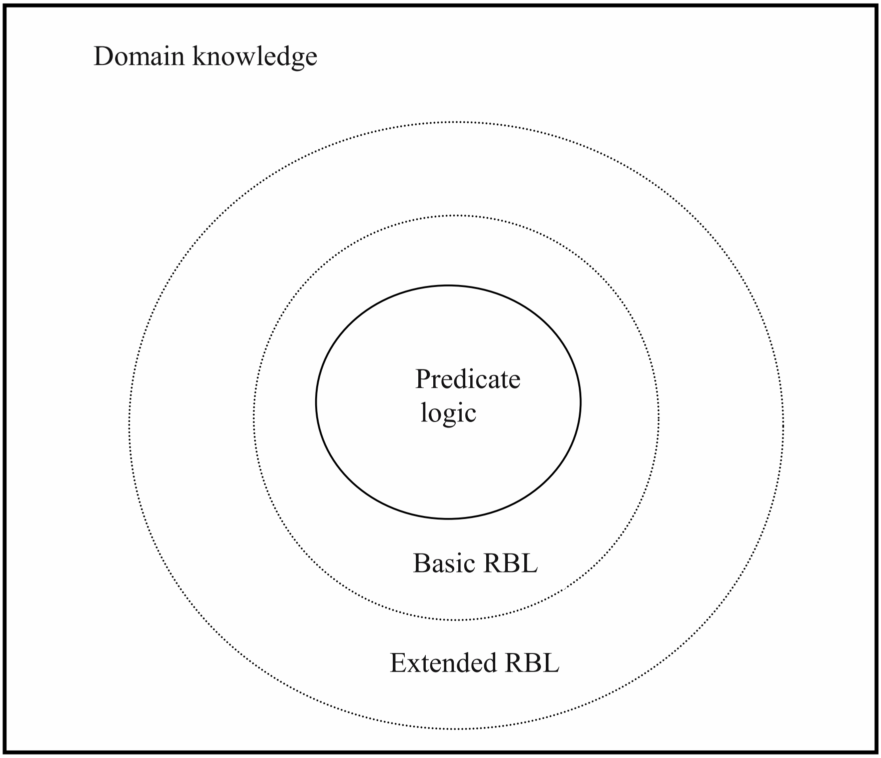
that there is no sharp boundary between logic and domain theory[3] and that a logic can be extended or limited, according to one's needs. RBL is in this view an extension of predicate logic that can be used for special reasoning tasks, in particular tasks in the fields of moral and legal reasoning. Moreover, according to this same philosophy, RBL can naturally be extended to deal with kinds of reasoning that cannot well be handled by its more limited versions. This means that there is no canonical version of RBL. The central function of RBL is to deal with reasons and their logic, but otherwise the logic can be limited to a core or extended to an apparatus with baroque pretensions. According to this same philosophy, logics are to some extent determined by the domains in which they are used, which means that their nature is to some extent established on the basis of empirical research. Logic is not completely a priori and logical theories are subject to changes that result from new insights, although not to the same degree as empirical theories.[4] The logic presented in this chapter is a relatively limited version of RBL that deals with contributive reasons and rules, but not with, for instance, goals and deontic predicates. In chapter 4 the logic is elaborated to deal with the comparison of alternatives.

The picture below represents the relation of the different versions of RBL to, on the one hand, predicate logic and, on the other hand, domain knowledge. The borderlines between basic RBL, an extended version of RBL, and domain knowledge are dotted to indicate that they are not very sharp.

In *Reasoning with Rules*, RBL was presented as a non-monotonic logic. However, I have come to think that the non-monotonic aspect of RBL is less central to it than I originally thought it was. Therefore I will deviate from my earlier approach here and mainly discuss a monotonic version of RBL. First I will present the language of RBL, by adding some extensions to the language of predicate logic and formulate a number of axioms that come on top of those of predicate logic. These axioms describe the logical behavior of the extensions to the language. There is no need for special RBL inference rules, because the inference rules of predicate logic suffice for the monotonic version of RBL.

[3] This theme is elaborated in Hage 2001 (LL).

[4] Partly for this reason, the version of RBL developed here differs in a number of aspects from the versions developed in earlier work, in particular in Hage 1996 and Hage 1997 (RwR), which - by the way – also differed amongst each others. The different versions of RBL reflect (slightly) different views of the logic of legal reasoning.



There will be no 'real' inference rules for the non-monotonic version of RBL. The idea behind the introduction of inference rules is that they allow constructive proof steps that lead from a set of premises to valid consequences from these premises. If a logic is non-monotonic, this constructive approach does not work, because non-monotonic logics base the logical consequences of a theory on the theory as a whole and what follows from it (exceptions should not be *derivable*). This means that the very idea behind the use of inference rules does not work for a non-monotonic logic, unless the purpose of derivation is changed. If the conclusions from a set of premises need not be true anymore given the truth of the premises, there is room for constructive inferences rules. However, the application of these rules does not lead to necessarily true conclusions, but to conclusions that are justified relative to the premises.[5] In section 7 I will give an informal indication of the different ways in which justification and justification defeat can be modeled.

[5] See chapter 1, section 2.3.

I will not describe a formal semantics for RBL. The semantics for the monotonic version of RBL would be the semantics of predicate logic with the additional constraint on the interpretation function that all the axioms of RBL (which define the special language elements) should be true in all models.[6]

In the following sections, I will first discuss a basic version of RBL (sections 2 to 4). Then follows an important extension to this basic version, namely a way in which RBL can deal with (legal) rules. In an appendix I will indicate the main differences between the present version of RBL and the version described in *Reasoning with Rules*.

2. THE LANGUAGE AND ONTOLOGY OF RBL

One important way in which RBL is an extension of predicate logic is that its language is an extension of the language of predicate logic. The extension consists mainly in a number of dedicated predicates, relations and function expressions that play a logical role in RBL. In this and the following sections, these extensions are introduced in an explanatory context. The first conventions concern a specification of the language for predicate logic that will be used:

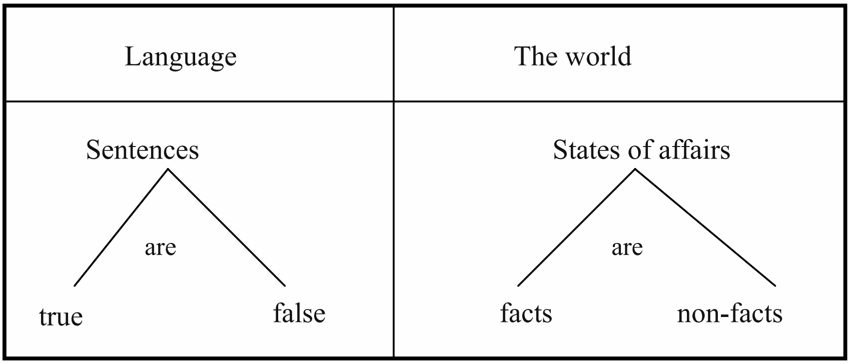
- All constants for relations, predicates and sentences without a subject-predicate structure start with an uppercase letter.
- All function expressions, individual constants and variables start with a lowercase letter, except individual constants and variables that denote or stand for states of affairs. These start with an asterisk (*), followed by a lowercase letter.
- The constants ∀, ∃, ~, &, ∨, → and ≡ stand for the universal and the existential quantifier, negation, conjunction, inclusive disjunction, the material conditional and equivalence, respectively.
- Variables are *italicized*.

2.1 Sentences, states of affairs and facts

RBL presupposes a rich ontology. Next to the 'ordinary' physical things, it also assumes several kinds of immaterial entities, including states of affairs and sets of individuals (in particular sets of reasons). States of affairs are

[6] I have presented a semantics for RBL along these lines in Hage 1997 (RwR), 223f. See also chapter 5 in which model theoretic semantics is given for rule logic.

what is expressed by sentences with truth values.[7] For instance, the sentence 'It's raining' expresses the state of affairs that it is raining. Some states of affairs *obtain* in the world; these are called *facts*. A sentence that expresses a fact is *true*. False sentences express *non-facts*, namely states of affairs that do not obtain.



In most declarative sentences it is possible to distinguish one or more terms that denote entities in the world. Next to these terms there will be a predicate expression by means of which something is said about the denoted entities. For instance, in the sentence 'John walks' the word 'John' denotes John and the word 'walks' is used to say something about John. In the sentence 'Jane gave the book to the father of Mary' the expressions 'Jane', 'the book', 'Mary' and 'the father of Mary' denote, while 'gave … to' indicates the relation between the three denoted entities. The expression 'the father of Mary' is a so-called function expression. It denotes the father of Mary, but it also contains the term 'Mary', which denotes Mary.

Logicians call the entities about which a sentence is (logical) *individuals* and the expressions used to refer to them *terms*. Terms should be distinguished from full sentences. Sentences have truth values, terms not. Even function expressions, although they contain a reference to an individual, have no truth values. The reason for this is clear: function expressions denote individuals; they do not state anything. So, there is, from a logical point of view, a fundamental difference between sentences and

[7] The clause 'with truth values' is meant to exclude non-descriptive sentences, such as commands, but also descriptive sentences that have terms on referential positions that have no object of reference, such as 'The king of France is bald'. Cf. Strawson 1950.

terms. Sentences have truth values; they do not denote.[8] Terms, on the contrary, have no truth values, but denote.

By assuming the existence of states of affairs, this clear distinction between on the one hand terms and individuals and on the other hand sentences and truth values, is blurred somewhat. Sentences no longer only have truth values, they can also be treated as terms that denote the states of affairs expressed by them. This happens, for instance, in sentences that deal with so-called propositional attitudes, such as 'Mary beliefs that John walks'. Taken by itself, the sentence 'John walks' has, in its quality of a sentence, a truth value, but as content of a propositional attitude it denotes the state of affairs that John walks.[9]

In RBL the distinction between the two functions of sentences is made explicit by syntactical means. The state of affairs expressed by sentence `S` is *typically* denoted by the term `*s`. In this way, a term that typically denotes a state of affairs indicates by its internal structure which state of affairs it denotes. Since states of affairs are logical individuals, they can also, non-typically, be denoted by other terms. For instance, the state of affairs `*it's_raining` can also be denoted by the term `*a`. In that case the sentence `*a = *it's_raining` is true.[10] In general the following translation holds between sentences and the terms that typically denote the states of affairs expressed by these sentences:

- If `S` is a sentence and if `s` is the string that results if all the uppercase letters at the beginnings of the atomic sentences that are part of `S` are replaced by lowercase sentences, then `*s` typically denotes the state of affairs expressed by `S`.
- If `*s` is a term typically denoting a state of affairs and `S` is the sentence that results if all the lowercase letters at the beginnings of terms denoting atomic states of affairs are replaced by uppercase letters, then `S` expresses the state of affairs denoted by `*s`.

[8] Frege, however, assumed that sentences denoted truth values. Cf. Geach and Black 1980, 62f.

[9] There are lots of complications here. For instance, the sentence 'Mary beliefs that John walks' might be interpreted as expressing a three-place relation between Mary, John and walking, rather than as a two-place relation between Mary and the state of affairs that John walks. See in this connection Quine 1956. For the present purposes I only assume that it is sometimes useful to treat sentences as denoting states of affairs and that sentences in their function of terms should syntactically be distinguished both from sentences in their function of expressing states of affairs and from terms which do not express states of affairs.

[10] The convention that terms denoting states of affairs start with an asterisk is also used for terms and variables that non-typically denote states of affairs.

Some examples:

- If It's_raining expresses that it is raining, then *it's_raining denotes the state of affairs that it is raining.
- *it's_raining & there_is_a_storm denotes the (compound) state of affairs that it is raining and there is a storm.
- *gives(john, mary, wedding_ring) denotes the state of affairs that John gives Mary the wedding ring.
- *∀*x*(thief(*x*) → punishable(*x*)) denotes the state of affairs that all thieves are punishable.
- *obtains_longer(*age(john, 6), *age(kim, 6)) denotes the state of affairs that John has been 6 years old during a longer time than Kim has been 6 years old. Notice that this state of affairs is about other states of affairs, which is reflected in the re-occurrence of asterisks in the term denoting the state of affairs.
- If *rescued(tarzan, father_of(jane)) denotes the state of affairs that Tarzan rescued Jane's father, then Rescued(tarzan, father_of(jane)) expresses this same state of affairs. Notice that the first letter of the function constant father_of remains lowercase in the sentence, because it is part of a term.

Variables for states of affairs start with an asterisk too. For instance, the following sentence expresses that Jane believes everything that John believes:

∀**s*(Believes(john, **s*) → Believes(jane, **s*))

If a sentence is true, the state of affairs expressed by it obtains.[11] RBL has in this connection a dedicated predicate constant Obtains/1, which operates on terms that denote states of affairs. The relation between the truth of a sentence and the state of affairs expressed by this sentence is rendered by the following axiom of RBL[12]:

Definition obtains:

∀**s*(Obtains(**s*)) ≡ S

[11] Notice that the obtaining of a state of affairs is not identical to its existence. The point of having states of affairs next to facts is that it is possible for a state of affairs not to obtain. The state of affairs that does not obtain must 'exist', because otherwise the statement that a particular state of affairs does not obtain would have a non-referring subject term. Those who object against this extended use of the notion of existence may consider to replace this notion in connection with non-obtaining states of affairs with the Meinongian notion of subsistence. Cf. Lambert 1995.

[12] This definition presupposes that the state of affairs *s is typically denoted by the term '*s'.

2.2 Abstract states of affairs

States of affairs are either abstract or concrete. An abstract state of affairs can be realized (instantiated) in different ways. For instance, the abstract state of affairs that somebody gives something to somebody else is realized by the concrete state of affairs that John gives Mary a book, but also by the concrete state of affairs that Russia gives the Netherlands a collection of drawings.

Abstract states of affairs are denoted by a term for a state of affairs that contains at least one free variable. For instance:

- `*rescued(tarzan, y)` denotes the abstract state of affairs that Tarzan rescued somebody. Notice that this expression is a term that denotes a state of affairs. In particular it should be distinguished from the sentence `(∃y)Rescued(tarzan, y)`, which expresses (rather than denotes) the concrete state of affairs that there is a person that Tarzan rescued and from the term `*(∃y)Rescued(tarzan, y)`, which denotes this last concrete state of affairs.
- `*gives(x, a_book, y) & (x ≠ y)` denotes the abstract state of affairs that somebody gives a book to somebody else.

Concrete states of affairs can instantiate abstract ones. A concrete state of affairs `*s` instantiates an abstract state of affairs `*s'`, if and only if there is some substitution i such that the term that typically denotes `*s` is the result of uniformly substituting all variables in the term that typically denotes `*s'` by constants according to i.

In this connection the function `instantiation/2` is relevant. The first parameter of this function is an abstract state of affairs and the second an instantiation. Its value is the concrete state of affairs that results from replacing all free variables in the first parameter by constants according to the instantiation of the second parameter. For instance:

```
*s = instantiation(*s', i)
```

Whereas states of affairs can be both abstract and concrete, facts are always concrete.