INTRODUCTORY REMARKS

This thesis is divided into two distinct parts. Part I proposes an approach for formal and semiformal representation of informal theories that have certain properties often prevalent in realistic knowledge. Part II gives an analysis preparatory to formalization of the selected domain which is legal knowledge. The result is the informal theory used to exemplify the representation approach proposed in Part I. After Part II the complete thesis is summed up in concluding remarks.

Multilayered domain knowledge, which is vague and fragmentary and therefore not fully formalizable, can be reproduced in a semiformal interactive metalogic program. This is the insight underlying the representation approach presented in Part I. The main topics of Part I are formalization methodology, representations of the domain knowledge by means of pure metalogic programs interpretable as first-order theories, and incorporating user interaction and coping with changes without distorting this declarative semantics.

Preparatory to formalization the informal knowledge must be studied, those components to be regimented within it identified, their interrelationships analysed and the structure of the informal theory reconstructed. Moreover, those properties of the informal theory that determine or influence what properties the selected formal language should have must be identified. Semantics of the informal knowledge should correspond to, or at least be consonant with, the formal semantics. The study of these aspects of the informal knowledge is carried out in Part II of this thesis.

We have attempted to make both Part I and Part II self-contained. A reader who is interested only in the formal representation of multilayered and vague knowledge can omit Part II of this thesis. Part I therefore includes a condensed account of the informal theory which is the result of the analysis in Part II.

This arrangement is intended to help readers with a background of either computing science or law, but not of both. The mutual independence of the two parts is reflected in the order between them. Though Part II is preparatory analytics, the informal theory resulting from this phase can be taken as a postulated theory for the analytics in Part I, just as we undertake formal metalogic investigations without taking a stand with respect to the multitude of controversial issues within the philosophy of logics.

PART I

SEMIFORMAL METALOGIC MULTILAYERED PROGRAMMING

Part I presents an approach for formally representing realistic domain knowledge. Characterized by properties often rendering knowledge intractable for formalization, the selected domain is a valuable object of study for knowledge representation. In particular, only a vague conception exists of the structure and content of its rules and facts and in addition the knowledge is multilayered. The domain is legal knowledge described in an informal theory where schematic descriptions of legal interpretation rules play a central role.

The investigation is devoted to computing science and confined therefore to the problems of representing the informal theory as a formal theory which is adequate modulo the informal theory and formally tractable. It is claimed that the informal theory is adequate modulo actual legal knowledge, but consonant with the purpose of Part I the argument supporting this claim is presented in a condensed form, and the more thorough motivation postponed to Part II.

In contrast to a single fixed interpretation, our representation incorporates multiple interpretations of legal sources. Briefly, our idea is to dynamically propose and revise formal theories, each of which represents one possible interpretation of the informal theory. These theories are formal object theories relative an interactive semiformal metalogic program which composes and assesses them in co-operation with the user.

Unrestricted multiple interpretation of knowledge implies that the knowledge has an arbitrary multilayering, since the canons of interpretation must be multiply interpreted in their own right, etc. It is shown that reasoning with such knowledge relies on an informal counterpart to upward reflection in that a sentence is legally valid if this is established by the canons of interpretation at the adjacent higher level. Our formal reasoning scheme for dealing with this is reminiscent of the so-called 'implicit reflection' in Reflective Prolog.

We present a novel knowledge representation methodology. Typically, formalizing involves three distinct theories: the informal theory to be formalized, the formal theory, and the informal metatheory discussing the latter, but for multilayered, imprecise informal theories we propose instead a semiformal metatheory discussing both theories.

Our semiformal metatheory is an interactive program. Illustrations are included for how sessions are composed, first, for computing the formal theory proposed as a representation of one possible interpretation of the informal theory and, second, for assessing the proof of this formal theory. Knowledge assimilation is accomplished without distorting the declarative semantics of the first-order formal theory.

Part I presupposes that the reader is familiar with mathematical logic, logic programming and metalogic programming.

CHAPTER 1

INTRODUCTION

In this chapter the methodology is introduced which we propose for coping with problems of formally representing multilayered fragmentary knowledge.

1.1. PURPOSE

Very seldom does a realistic domain consist of precise knowledge. Instead, most often only an abstract conception of its rules and facts exists. To determine their content exactly, once and for all, is simply not feasible. From such fragmentary knowledge well-founded conclusions cannot be reached with which to resolve problems properly. Therefore a preparatory and major task in the reasoning process is to reduce lacunae ('gaps') in the knowledge available and hence at least suggest a basis from which decisions can be reached. This task is perhaps accomplished in a capricious manner in some domains, but in others instead is carried out in accordance with established 'interpretation principles'. Legal knowledge, for instance, could be claimed to form a multilayered hierarchy of fragmentary descriptions of such principles. The object of study in this thesis is an informal theory describing legal knowledge in this manner.

We have chosen this object of study because legal knowledge exhibits a clearly discernible layering. The levels have been detected as a result of lengthy analyses in legal philosophy and practice, a work which spans over at least two and a half millennia, but nothing suggests that the knowledge structure is distinctive for the legal domain. On the contrary, analogous layering and problems of fragmentation can be discerned, for instance, in the knowledge of a domain of malfunctions in human beings, cf. e.g., Sticklen *et al.* [69] concerning medical diagnostic reasoning. Hence, we assume that good solutions to the representation problems in the domain of legal knowledge will have general applicability.

The purpose of this thesis is to investigate the prospects of giving an executable formalization of informal theories which describe multilayered fragmentary knowledge. As a representation for such an informal theory we propose a formal object theory and specify how it can be defined in an interactive semiformal metatheory encoding rules for reasoning about (i) the assimilation of external knowledge into the formal object theory, and (ii) the existence in it of formal proofs and their structure. Statements in the semiformal metalanguage of the metatheory link symbols of the formal object theory to their intended interpretation in the informal theory. Both the semiformal metatheory and the formal object theory can be interpreted as first-order theories although the implementation of the former presently relies on a few imperative constructs.

1.2. BACKGROUND

Metaprogramming is an important technique for the three interrelated topics 'knowledge representation', 'knowledge processing', and 'knowledge assimilation' (Kowalski [50]). The first deals with the apt choices of formalism and approach to building formal theories, the second with the construction of proofs of theorems from such theories and/or the identification of the existence of such proofs, and the third with the assimilation of new knowledge into existing theories. They are all interrelated insofar as the formalism and approach for representing knowledge are largely determinative for the choice of method for constructing proofs, methods which in turn are exploited for analysing new knowledge proposed for assimilation into existing theories. In this thesis it is described how these three topics and their interrelationships are involved in a semiformal metatheory characterizing fragmentary, multilayered, not fully formalizable legal knowledge as a metaprogram interactively constructing metaproofs. The implementation of the metaprogram facilitated the study of the complexity of the three topics and their interrelationships as well as the adequacy of the formalization attempt. Our metaprogramming approach is untraditional in that the technique is used mainly for construction of object theories, not so much for control. Consonant with this objective, our naming convention takes structural-descriptive names as compound entities that can be taken apart and put together to form new names, thus describing new expressions of a theory.

This study shows how *upward reflection* can be used as a powerful reasoning method. Deriving its origin from Feferman's [19] work in mathematical logic in the early 1960s, reflection was introduced in the early 1980s in mechanized formal reasoning by Weyrauch [77] and in metalogic programming by Bowen and Kowalski [11]. As a theoretical construction the notion of reflection has thus been around for some time in computing science. Its potentials have been demonstrated in artificial domains, e.g., for representing multiagent belief and knowledge [50] and for exploiting properties of predicates such as symmetry [16]. To the best of our knowledge, however, our study is the first to illustrate reasoning with realistic knowledge which requires upward reflection. Therefore, we hope it will contribute to the understanding of reflection as a knowledge representation tool. For instance, it indicates that, while upward reflection has its informal counterpart in legal reasoning, the application of downward reflection on the contrary violates the inherent structure of legal knowledge. The counterpart to upward reflection in legal reasoning is connected to 'rules of legal interpretation' and can be concisely described. If we propose a legal rule for solving a legal case we must show that the rule's structure and content are in accordance with the (meta)rules of legal interpretation, otherwise the rule is legally invalid. Likewise in automatized legal reasoning, a formula A representing a legal rule can be assumed included in an object level theory OT representing legally valid rules if its inclusion accords with the metalevel theory MT of formulas representing rules of legal interpretation, i.e., assuming $Demo$ defines provability, we have $Demo(name(OT), name(A)) \leftarrow Demo(name(MT), name(Demo(name(OT), name(A))))$ where $Demo$ holds for formulas belonging to or deducible from a theory, cf. Kowalski [50]).

Though inessential in principle, metaprogramming is often convenient in practice. Reasons may be its naturalness of representing the domain knowledge or even the impracticability of giving perfect object level representations. A clear parallel as to legal knowledge has been described by Horovitz ([44], p. 94), in a deservingly concise comment on the role of 'rules of legal interpretation': "...they are inessential in principle, in the sense that, although they are necessitated in practice by the imperfections and the dynamic character of the existing systems, they would not be needed in a perfect, unambiguously formulated, consistent, and complete legal system, conformable to a stable social reality. The actual function of rules of legal interpretation is to direct the identification of the existing system and its continuous construction and readjustment." Horovitz' observation seems inescapably true. Realistic systems of legal rules are always subjected to imperfection, ambiguity, etc., which necessitate metasystems of (meta)rules of legal interpretation. However, an equally inescapable truth is that such metasystems are also, in turn, subjected to imperfection, etc., necessitating metametasystems, etc.

Consider Horovitz' unattainable 'perfect' legal system and try to imagine its 'ideal' complete formalization. Two equivalent perspectives appear, the first excluding and the second including the role of the rules of legal interpretation. An ideal formalization corresponding to the first perspective would be a potentially infinite object theory explicitly embracing all valid object theory legal rules. An ideal formalization corresponding to the second perspective would be a multilevel theory structure implicitly embracing all valid object theory rules because these can be established by repeated application of all valid metatheory rules (metarules of legal interpretation) which in turn can be established by repeated application of all valid metametatheory rules (metametarules of legal interpretation) which in turn can be ...established by repeated application of all valid meta...metarules of some topmost theory which in this ideal case is assumed to be fully specified. Incorporating rules of higher levels in the first perspective yields a third perspective where higher-level rules can be understood as a complete characterization of the explicit object theory in another language or languages.

Of course, an unattainable ideal formalization can only be axiomatized partially and approximately. The extent to which the informal legal theory can be reproduced in this axiomatization, however, seems to depend on which of the two perspectives is adopted. The explicit single-level object theory of the first perspective would yield a static formalization consisting of an enumeration of non-logical axioms representing suggestions for object theory rules. In contrast, the multilevel theory structure of the second perspective would yield a dynamic formalization in which generation of object theory rules is reflected. Thus, from a practical point of view this latter alternative seems to have the best potentials and it has therefore been our guiding star. That is, we have attempted to realize in a metalanguage a partial axiomatization of a multilevel theory structure roughly corresponding to the second perspective.

1.3. METHODOLOGY

We have found the following notions useful for analysing formalization of knowledge with multilayered structure. Kleene ([49], pp. 65, 69) introduces three separate and distinct 'theories' involved in the process of a formalization:

(a) the informal theory of which the formal system constitutes a formalization,

(b) the formal system or object theory, and

(c) the metatheory, in which the formal system is described and studied

where theory (b), which is formal, is not a theory in the common sense, but a system of symbols and of objects built from symbols described from (c). Theories (a) and (c), which are informal, do not have an exactly determined structure, as does (b).

Consider the following two approaches for studying theory (b).

(i) the formal theory (b) is "introduced at once in its full-fledged complexity" and investigated by methods without making use of an interpretation. (This is known as the metamathematical approach if the methods are finitary.)

(ii) the formal theory (b) is studied by recognizing an interpretation of the theory under which it constitutes a formalization of (a), i.e., we analyse existing informal theories (a), "selecting and stereotyping fundamental concepts, presuppositions and deductive connections, and thus eventually arrive at a formal system", i.e., at the formal theory (b).

Approach (i) presupposes that the complexity of the formal theory is fully understood, which, for obvious reasons, it is not in our domain. A realistic system can only have a partial axiomatization of the formal object theory. This axiomatization can be gradually extended, though, by consulting the user both for supply of metalinguistic entities representing objects of the formal object theory and for completing formal proofs in it. Thus, in a practical system we must adhere to approach (ii). However, for a preparatory isolated study of the two topics knowledge processing and knowledge representation, it may be more elucidative to make the necessary adjustments to carry out the study along approach (i).

Various types of legal knowledge raise their particular representation problems and call for different formalization approaches. In this thesis the following knowledge categories are touched upon.

- rule-like expressions (such as statutory rules, other provisions, legal interpretation principles, etc.) which mediate schematic descriptions of rules for resolving legal cases,
- descriptions of situations (precedent legal cases), by interpretation considered subsumed by these schematic rule descriptions, and afterwards serving as their interpretation data,
- descriptions of situations (precedent legal cases) from which 'rules' are induced to solve unregulated cases,
- incomplete and imprecise descriptions of the current case, and
- the problem which of these four should constitute formalized or external knowledge, and how the latter should be supplied and assimilated.

Demarcating a study so as to exclude the last of these items makes possible an investigation along approach (i). We have carried out such a study [34] in which we devised a Horn clause metalanguage axiomatization of a formal object theory (b) intended for formalization of hierarchical fragmentary knowledge. The language of this formal object theory is an n level language where each level $i+1$ is the formal metalanguage of the language of level i. Our intention was to study knowledge processing and representation within the formal object theory itself. We considered it appropriate to start studying these two subjects isolated from the problems of supplying and assimilating external knowledge. We carried out the investigation as though we had had an ideal one-to-one axiomatization of the underlying formal object theory (b). Such an axiomatization is necessarily ideal because it is impossible to anticipate all knowledge fragments needed to complete the computation for a 'real life' application. They must be supplied from the outside. However, an investigation can be carried out as though a one-to-one axiomatization were available by simply assuming, as we did, that sufficient fragments are available for the particular cases studied.

In subsequent studies [33,35] we delved into the knowledge assimilation problem 5, which made it appropriate to study the formal object theory along approach (ii). We removed the simplifying assumption that a full metalanguage axiomatization of the formal object theory is obtainable and hence also the possibility to predetermine sufficient knowledge fragments of its various levels. Although the non-logical axioms of the formal object theory (b) cannot be enumerated in advance, its possible content nevertheless can be discussed in a theory (c) of a metalanguage which may be informal—but also formal, or both. To this end we devised a semiformal metalanguage for a theory (c) which axiomatizes the 'available' part of the formal object theory and encodes rules for the assimilation into it of externally supplied knowledge fragments. The semiformal metalanguage has as object language the n level language of the formal object theory. Knowledge assimilation is dependent on the deductive structure of the formal object theory. This can be accounted for in the semiformal metalanguage since its objects of discourse include formal proofs, i.e., sequences of formulas of the formal object theory. Below, $\mathcal{IT}$, $\mathcal{OT}$ and $\mathcal{MT}$ denote, respectively the (a), (b) and (c) of our system.

1.4. HISTORICAL REMARKS

Our work on representing the multilayered structure of legal knowledge was initiated in 1988 and the first implementation presented in 1989 and in 1990, Hamfelt & Barklund [31] and [32]. We first described $\mathcal{OT}$ in a thesis in 1990 ([29], pp. 27–35). In subsequent studies, Hamfelt & Hansson [33,34,35], $\mathcal{MT}$ was distinguished from $\mathcal{OT}$ and the implementation of $\mathcal{MT}$ as an executable interactive Horn clause program carried out. At the same time, Barklund & Hamfelt [2] carried out a study of a formal metalanguage incorporating logical components above those available in Horn clause logic for representing legal rules, such as classical (but somewhat incomplete) negation and equivalence.

1.5. DEMARCATION

Part I of the thesis is a study of the feasibility of formalizing a certain informal theory, not a study *per se* of the adequacy of this informal theory modulo what it purports to describe. In our opinion, it is important to maintain this demarcation in a computing science thesis, since the issue of adequacy belongs, not to computing science, but to legal philosophy with its multitude of ramifications.

In this sense $\mathcal{IT}$ is a postulated theory, not an account of actual legal knowledge. This is not to say, however, that $\mathcal{IT}$ is not a good candidate for an adequate description, at least of certain important aspects of legal knowledge. It is, inasmuch as it has much in common with theories put forward by prominent legal theorists. In Part II of this thesis, support in legal philosophy is briefly summarized. Furthermore, we present a more thorough argument for how the concept of law is permeated by the principles of legal interpretation at their various levels, from the treatment of vague legal concepts and their semantics, to the more obvious role these principles play for particular legal inferences such as *analogia legis* and *e contrario*. Furthermore, we claim in defence of classical logic that (i) we strive to reproduce lawyers' understanding of legal rules, not the textual representation *per se* these rules have in a statute, (ii) this understanding is determined by the lawyer's function, (iii) a major function of the vast majority of lawyers, such as attorneys and judges, is to apply legal rules to factual situations, and (iv) this group therefore construe the rules as indicative sentences, while law makers and their addressees perhaps construe them as sentences of the imperative mood or containing deontic modalities.

Of course, as all informal theories intended to be formalized, $\mathcal{IT}$ has been subjected to a regimentation. Some aspects of real legal knowledge have been brought forward and elucidated, while others have been disregarded.

1.6. OUTLINE

In the sequel we first give an abstract characterization of the multilayered informal legal theory $\mathcal{IT}$. Then a theoretical analysis follows of the formalization of $\mathcal{IT}$ as a multilevel formal object theory $\mathcal{OT}$ characterized in a semiformal metatheory $\mathcal{MT}$. Afterwards, the actual implementation of $\mathcal{MT}$ is scrutinized, beginning with a code level description of the central part of the program—an interactive semiformal theorem prover—and then illustrating its execution by tracing the computation of a sample query. The need is stressed for interactive assessment of the reasoning leading up to a conclusion and a solution is presented in which the proof term representing this reasoning is unfolded piecemeal in parts convenient for display. Next, appropriate 'query the user facilities' are sketched for a system of this kind whereupon a session is listed that is intended to illustrate how the system interacts with the user. It is argued that $\mathcal{MT}$ is highly modular, enabling one to cope with changes in the frequently modified legal knowledge it represents. Part I of the thesis ends with an account of related work before conclusions and further work.

TYPOGRAPHICAL CONVENTIONS FOR PART I

Formulas in typewriter font	Implemented logic programs
Formulas in italics	Theoretical logic programs *or* regimented characterizations of informal theories
$\forall$x, $\leftarrow$, $\wedge$	Symbols that the theorem prover processing the formalism interprets as universal quantification, implication, and conjunction
Words in roman font within italic formulas	Symbols interpreted as variables by the theorem prover processing the formalism
$(x)[\ldots]$, $\Leftarrow$, $and(\ldots,\ldots)$	Term level descriptions of universal quantification, implication, and conjunction; these are meaningless to the theorem prover processing the formalism
`<=`, `and(...,...)`	Term level descriptions in implemented logic programs
$n(\ldots)$, `n(...)`	Metalevel names for expressions

CHAPTER 2

THE INFORMAL LEGAL THEORY

In order to motivate the approach chosen we shall describe in detail our conception of the structure of legal knowledge in this chapter, i.e., our informal theory $\mathcal{IT}$*. The purpose is first to convey an understanding of how rules at different levels operate together and second to propose an informal characterization of* $\mathcal{IT}$*. This characterization will then serve as a basis for the formalization developed in subsequent chapters.*

2.1. LAYERS OF LEGAL KNOWLEDGE

We take as point of departure an informal theory $\mathcal{IT}$ describing legal knowledge as a multilayered hierarchy of vague rule descriptions from which proper rules may be proposed for resolving particular cases. Fig. 2.1 illustrates a conceivable fragment of the hierarchy of $\mathcal{IT}$.

Consider provision 1 in Fig. 2.1, an ordinary statutory rule from the Swedish Sale of Goods Act (Sect. 5, SGA). This provision is applicable not only to sale of goods. It could, e.g., be analogically applied to, say, hire of goods, or extensively interpreted, or interpreted by inversion (*e contrario*), etc., and thus embraces numerous primary rules. One and only one of these is the rule given by a literal reading of the tokens building the provision and not even this rule has a legal validity which can be taken for granted. Provision 1 is a *schema* for all these rules and since this schema is about primary rules and thus conceptually belongs to the secondary level we call it a secondary schema.

The relation between secondary schemata and primary rules is given by secondary rules. For example, the relation between schema 1 and real primary rules is given by secondary rules such as rule 2 in Fig. 2.1 which is just an example of how a secondary rule for *analogia legis* in commercial law could possibly look. In the same way there exist tertiary schemata for secondary rules and tertiary rules that give the relation between these schemata and the secondary rules, etc. Secondary rule 2 originates from tertiary schema 3. Information about the relation between this schema and secondary rules such as rule 2 is given by tertiary rules, such as rule 4. Another tertiary rule 5 could be "In commercial law *analogia legis* may not be applied in a way which counteracts free competition".

Schematic descriptions of rules at various levels are important. We have argued elsewhere, cf. Part II and [29], that a lawyer has only a schematic knowledge of legal rules; each adjudication comprises an interpretation of schemata for

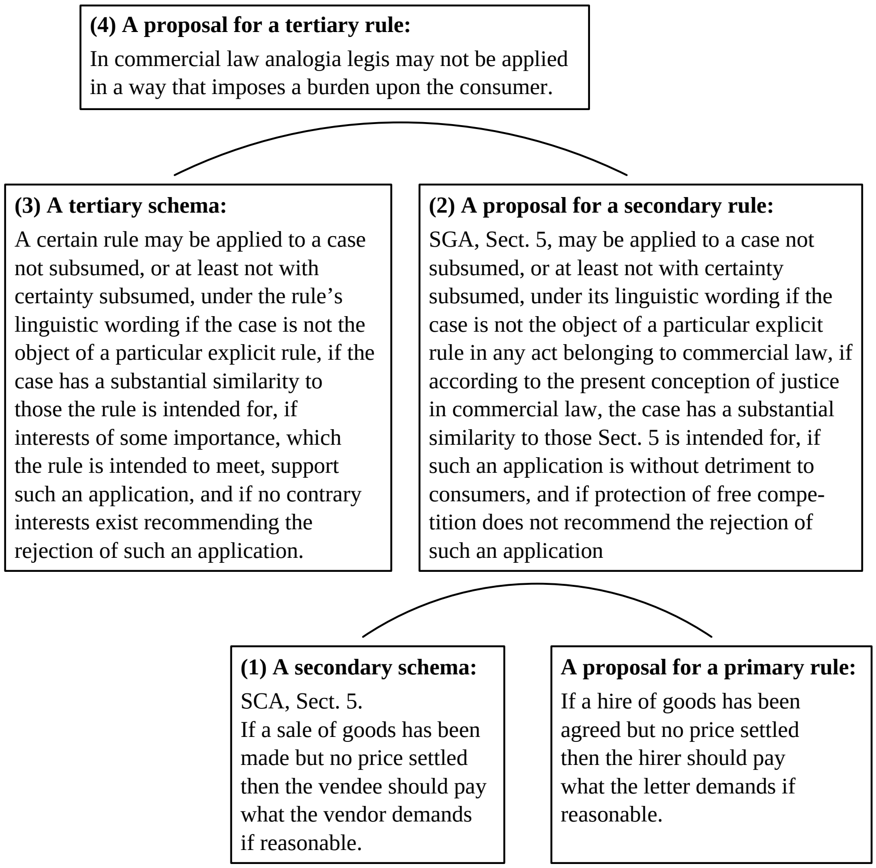

Fig. 2.1. Schemata and rules.

legal rules and results in the construction of specialized rules applicable solely to the case at hand. An obvious example is our secondary rule 2 for *analogia legis* in commercial law. It is not generally applicable. The rule is the result of an interpretation at levels above the secondary and only 'applicable' in an individual adjudication, i.e., in a particular legal case. In another legal case the interpretation at the levels above the secondary may yield another formulation of rule 2. Rules, such as rule 2, are thus generated for each individual adjudication and there is a variety of possible formulations. What they have in common is that they all originate from a common schematic description 3 which in this case belongs conceptually to the tertiary level. The schematic description 3 originates from the legal literature ([70], p. 71). Thus, rules 4 and 5 are in turn possible specializations of quaternary schemata, proposed by quaternary rules which in turn are specializations of quinary schemata, etc.

Fig. 2.2 illustrates the hierarchy of legal knowledge in which the levels have been made distinct by introducing, at each level i, the *names* for the rules of the level $i-1$.

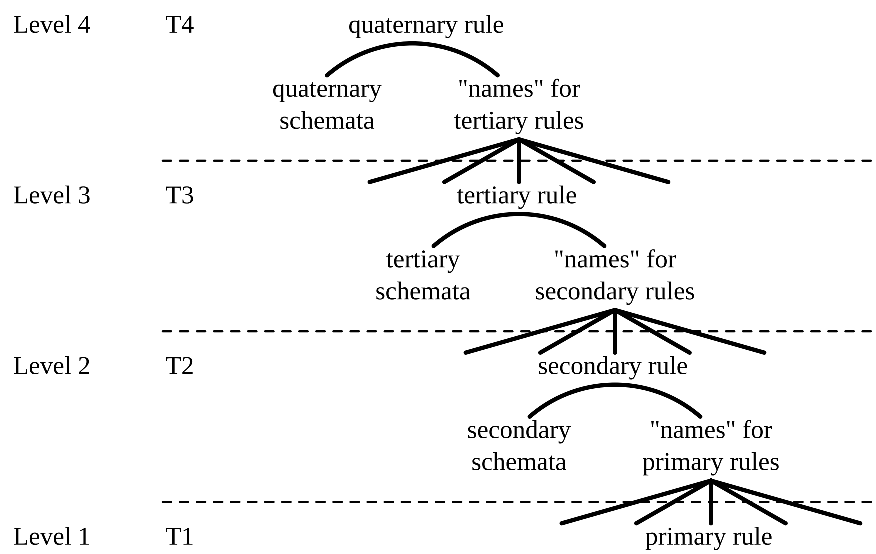

Fig. 2.2. Levels of legal knowledge.

The specialization of schemata (principles) must yield *meaningful* rules which are legally *acceptable* for the current case. The first means that the rule consists of nothing but legal concepts embraced by the principle, the second that its content is in addition legally adequate and its premises fulfilled so that it does in fact apply. The latter is settled recursively, i.e., a sentence is legally acceptable if either its content is directly accepted by rules at the higher adjacent level (base case), or it follows from other acceptable sentences at its own level (recursive case). The base case is thus informal upward reflection where the upper level enforces a sentence on a lower level.

Hitherto, rule specialization has been described in a top-down perspective. Specializations cannot, however, be carried out without interpretation data, the supply of which has character of a bottom-up process, since these data originate mainly from a description of the legal case. If C is a description in legal terminology of a legal case and J is a suggestion for a judicial decision, then $J \leftarrow C$ is a proposed primary rule. Establishing that $J \leftarrow C$ is accepted by the (secondary) metarules of legal interpretation then coincides with resolving the case. If the legal case is a precedent case $J \leftarrow C$ will be a proposed *ratio decidendi* of the precedent.

The account in this section should suffice as a basis for the computing science topics focused on in Part I of the thesis. A more thorough treatment of the multilayered structure of legal knowledge is found in our previous texts, in particular [29,34], see also [32]. In Part II of the present work, the interested reader will find references to theories in legal philosophy, which can be referred in support of our multilayered description of legal knowledge. Moreover, we illustrate a possible role for principles of legal interpretation for dealing with vague or open textured legal concepts, and in fact defining their semantics, to support a claim that the influence these principles exercise extends beyond such legal inferences as *analogia legis* and the like.

2.2. CHARACTERIZATION OF THE INFORMAL LEGAL THEORY

In the preceding section we described the particular conception of legal knowledge chosen for this study. Before a formalization can be proposed a more precise characterization of the informal legal theory $\mathcal{IT}$ is required and we suggest one such below.

Consider Fig. 2.2. Level 1 in the figure may be understood as an object level, level 2 as its metalevel, level 3 as its metametalevel whose object level is the pair level 1 and level 2, etc. The structure of the informal legal theory $\mathcal{IT}$ could be understood as a hierarchy of 'theories' where the language of each level i consists of all meaningful rules expressed by schemata and conditions for their specialization on the level $i+1$. Each T_i is characterized in T_{i+1} by schematic descriptions of its sentences and rules for deciding which specializations of these are meaningful and legally acceptable. The same holds for theory T_{i+1} with respect to theory T_{i+2}, etc., and since no level has rules which do not require interpretation, this proceeds *ad infinitum*. This endlessness can be dealt with by choosing some level n to be the 'topmost' at which discretion is used to specialize schemata, thus making the validity of a primary rule proposed for resolving a case and of all rules on each level i, $1 \leq i < n$, ultimately depending on discretion.

Thus, the informal legal theory $\mathcal{IT}$ can be characterized as a set of theories $\{T_1, T_2, \ldots, T_n\}$ where each theory T_i is understood as a collection of sentences fully determined by the higher adjacent theory T_{i+1}, i.e., if from theory T_{i+1} a theorem can be deduced expressing that a sentence A_i is provable from theory T_i, then A_i belongs to theory T_i. Therefore, we need a definition of the *provability relationship* between two adjacent levels $i+1$ and i ranging over n distinct levels (theories), and forming a hierarchy of dependent relations directed from the highest to the lowest level. On each level i this provability relation $T_i \vdash A_i$ should coincide with the rules of logic. The hierarchical dependence of the provability relationship, $1 \leq i < n$, may be characterized as follows

$$T_i \vdash A_i \quad \textit{iff} \quad T_{i+1} \vdash `T_i \vdash A_i',$$

where '$T_i \vdash A_i$' names $T_i \vdash A_i$. With theory T_n specified, the hierarchy decides the content of the object level as well as of all the other levels. That is to say, T_n decides the contents of all the theories $T_1, \ldots, T_{n-1}$. The definition constitutes the *rules of acceptance* of the informal legal theory $\mathcal{IT}$.

Lower levels are generated from higher adjacent levels by application of the *if* part of the above definition. This is an upward reflection rule allowing us to investigate the content of an object level theory by proving a metalevel sentence expressing something about this content. Upward reflection seems to have certain restriction in legal reasoning, however, depending on the lower level rules concerned. It seems quite conceivable that any sentence proposal for theory T_1, without prior inferencing at its own level, may be upward reflected to level 2, simply because in practice antecedents of primary rules most often refer to the facts of the case, rather than to consequents of other primary rules, and therefore long inference chains are rare at level 1. In contrast, such chains are not unusual for theories i, $i > 1$, which together with unconstrained upward reflection yields quite an unnatural reasoning. A rule A proposed for level i would be immediately upward reflected to level $i + 1$, at which a rule would be proposed for its assessment, but in turn this rule would be immediately upward reflected for assessment at level $i+2$, etc., forcing the reasoning process to ascend to the topmost level before taking the first inference step at level i.

A possible remedy is to constrain upward reflection at levels i, $i > 1$, to unconditional sentences (facts) which do not match any consequents of other rules at their own level. Conditional sentences (rules) on the other hand are assumed to be sentences of the current theory directly but do not apply before sufficient facts to satisfy their premises have been assessed and accepted by the theory of the higher adjacent level. According to this reasoning scheme the content of all sentences involved in the reasoning process will eventually be assessed. This facilitates the implementation of $\mathcal{IT}$ and probably in real legal reasoning something like this must be involved. We do not claim, however, that this particular restriction has any directly corresponding informal counterpart in real legal reasoning but nevertheless use it in our implementation.

Rules of acceptance of a theory T_{i+1} decide the content of theory T_i, i.e., they determine which are acceptable rules among the meaningful rules of level i. The meaningful rules constitutes the language of the level and are distinguished from meaningless expressions by *rules of meaningfulness*. Rules of meaningfulness appear at the metalevel with respect to a level i. They may be considered part of the rules of acceptance of level $i + 1$ or they may be considered part of a separate and, with respect to $\mathcal{IT}$, extra-systematic metatheory $\mathcal{MT}$.

CHAPTER 3

FORMALIZATION OF THE INFORMAL THEORY

This chapter discusses the properties of a formalization $\mathcal{OT}$ of the informal legal theory $\mathcal{IT}$ presented in the previous chapter. First, we identify certain basic problems that must be taken into account when developing $\mathcal{OT}$. Then a general representation approach is chosen for formalizing obtainable parts of $\mathcal{OT}$. $\mathcal{OT}$ must be a theory of a suitable formal language the properties of which we focus on next. Finally, we describe two different characterizations of the hierarchical dependencies in $\mathcal{IT}$ each of which serves as an alternative base for $\mathcal{OT}$. $\mathcal{IT}$ may be characterized as a single theory T_n, of an arbitrarily selected topmost level, and all the lower level theories $T_{n-1}, \ldots, T_1$ described in T_n. Alternatively, $\mathcal{IT}$ may be characterized with each of $T_n, \ldots, T_1$ as a separate theory, supplemented by a means of communication between these theories. Depending on this choice $\mathcal{OT}$ may be developed either in a single formal language in which the informal language of T_n is represented as formulas but $T_{n-1}, \ldots, T_1$ as terms, or in a composed formal language in which the informal languages of each of $T_n, \ldots, T_1$ are represented by formulas.

3.1. PRELIMINARIES

Certain problems must be taken into account in a formalization $\mathcal{OT}$ of $\mathcal{IT}$, both with respect to the representation of and the processing in the informal legal theory $\mathcal{IT}$. In this section we point out the problems we consider central.

In $\mathcal{IT}$ every legal schema has to be specialized before it can be applied as a rule to a specific legal case. It must be assessed how well each thus specialized rule accords with the legal principle it is intended to reflect. Some of the proposed specializations can be decided to be meaningless and therefore not adequate because some of the concepts used are not relevant legal concepts of the legal principle in question. For instance, it is meaningless to investigate whether a dog is allowed to inherit a human being. Thus, *meaningful* specializations should be sorted out from meaningless by typing conditions. Among the resulting meaningful rules *acceptable* legal rules are selected by legal principles of a higher level.

Thus, when developing a formalization $\mathcal{OT}$ of $\mathcal{IT}$ there are some problems in deciding the *formal non-logical axioms* of $\mathcal{OT}$. Often the main structure or form of a legal rule can be decided, but some parts of this structure are open for different legal concepts. The class of possible legal concepts in a given place in this structure cannot be decided in advance. Thus, there also are some problems in deciding the *formal syntax* of the formal theory, e.g., its primitive terms are not predetermined.

An example might elucidate these representation aspects. Recall the principle 3 of *analogia legis* in Fig. 2.1 (note that in the text there is an implicit reference to a legal field or setting to which the 'rule' belongs). Let us discuss a possible 'stereotyping' into a formal schema of this principle, which aspires to describe what is common for all secondary rules for *analogia legis* of all legal fields.

> A certain rule may be applied to a case not subsumed, or at least not with certainty subsumed, under the rule's linguistic wording, *if* the case is not the object of a particular explicit rule, *if* the case has a substantial similarity to those the rule is intended for, *if* interests of some importance, which the rule is intended to meet, support such an application, *and if* no contrary interests exist recommending the rejection of such an application.

An almost fixed logical structure (the *if*'s) can be identified in this principle. Roughly, the principle can be understood as a vaguely expressed 'definition', i.e., as an 'if and only if' statement with a 'definiendum' and its conditions (the 'definiens'). This is because it is intended to state when *analogia legis* can be applied, but also when it cannot. That is to say, for every legal case under consideration a court has to decide exactly whether or not *analogia legis* can be applied to the specific legal case, provided, of course, that *analogia legis* is relevant. The definiendum can be understood as a relation between a legal field, a legal provision, and a legal case. In the text the arguments of a relation are underlined; the other parts of the text characterize relations.

The relations can be regarded as having an almost *fixed structure*. However, a relation represents a vague legal concept with a totally *unspecified extension* which is only intuitively understood. Other legal principles decide whether or not a given specialization of a relation is an element of this extension. Only when a schema is completely specialized can it become a legal rule dedicated to a legal case and a provision of a legal field. This specialization comprises two phases. The rule has to be a meaningful rule. Moreover, a meaningful rule has to be deemed relevant to the current legal case. Afterwards, the acceptance of the conditions of the rule has to be determined.

With the presumption of an implicit equivalence, a natural way to represent this principle of *analogia legis* could be as a Horn clause, e.g., schematically described as

```
analogia_legis(n((Cons<=Ante)),n(Modifications),LegalSetting):-
  and(not(casuistical_interpretation(LegalField,
                                     n((not(Cons)<=Ante)))),
```

```
and(intended_for(ProvisionNo,n(TypeCase)),
and(substantial_similarity(n(TypeCase),
                           n(Ante),
                           n(Modifications)),
and(intended_to_meet(ProvisionNo,Interests,LegalField),
and(supports(ProvisionNo,n(Modifications),ProInt,Interests),
and(recommend_rejection(ProvisionNo,n(Modifications),
                        ContraInt,Interests),
```

where the head of the clause represents the application of the principle, and its body represents the conditions (premises) for this application. As mentioned, this principle (schema) is turned into a legal rule only when it is completely specialized, and judged meaningful and relevant. The clause could be read as: The principle of *analogia legis* allows a legal provision in a legal setting to be analogously applied to a legal case if the premises for this application are fulfilled, i.e., if the 'body' of the principle is found valid by other legal principles. More precisely, it says that the provision `(Cons<=Ante)` may be adapted with `Modifications` into an analogous rule which, as described on p. 14, represents a legal case and, in addition, since it has been found valid as well as applicable, resolves the case. It is worth noting that with the implicit assumption of an equivalence a completely specialized clause fulfils the demands on completion and groundness for an adequate application of the rule of 'negation as failure' [14].

3.2. REPRESENTATION APPROACH

How should we approach a representation of $\mathcal{IT}$? Since higher-level rules and schemata concern lower level rules and schemata, one could readily conceive of *metaprogramming* as a natural approach for their representation as well as for carrying out legal reasoning. Furthermore, the role played by the logic provability notion for the relation in $\mathcal{IT}$ between a theory T_i and its adjacent lower level theory T_{i-1} makes metaprogramming in *logic* a reasonable choice. For metaprogramming in logic, various languages have been devised, spanning from completely amalgamated languages where the metalanguage is identical with the object language, to completely separated metalanguage and object language. Among these we must choose an appropriate formal language for formalizing $\mathcal{IT}$.

We need the notion of *reflection* between object and metalevel theories, since upward reflection has an informal counterpart in legal reasoning. Reflection is a notion introduced by Feferman [19]. In his sense a procedure is a 'reflection principle' if it adds something to a theory which is unprovable within the theory itself but follows from the (material) 'validity' of the theory, e.g., a statement that the theory is consistent. Subsequently, various senses of the notion 'reflection principle' have evolved, deviating more or less from the original; for a review see e.g., Giunchiglia and Smaill [25]. In computing science, Weyhrauch [77] used the term for linking rules which allow an object level execution to be replaced by a metalevel execution, and vice versa. These rules could be called the upward and downward reflection rule, respectively. Bowen and Kowalski [11] have proposed a general scheme for metaprogramming in logic by using 'reflection principles' to amalgamate an object language L with a metalanguage M. In this thesis the term 'reflection principle' is used in its computing science sense.

Both the amalgamation and classical studies of the object-meta language partition—cf. e.g., Tarski's truth theory [74]—require that every term or formula E in the object language L be associated with a ground term $n(E)$, naming E, in the metalanguage M. Moreover, L forms part of M. This is repeated at the metametalevel and so forth. So far this proposal seems adequate vis-à-vis our understanding of the informal legal theory but we will have more to say about the appropriateness of the amalgamation later. We prefer to keep the levels distinct from each other, however. Schemata in $\mathcal{IT}$ must be specialized into proposed sentences before being applied in inferencing, which presupposes a preparatory meta-activity at the adjacent higher level.

A representation of $\mathcal{IT}$ must be reasonably intelligible. $\mathcal{IT}$ is a rather complicated informal theory and things would otherwise easily get out of hand. What makes $\mathcal{IT}$ so involved is that its topmost theory T_n completely determines the content of all lower level theories $T_{n-1}, \ldots, T_1$. At first sight, this seems to call for a *global* characterization of $\mathcal{IT}$. That is, all formulas and terms of theories $T_{n-1}, \ldots, T_1$ are represented in theory T_n whose language thus contains a representation of all the languages of these theories. We describe such a *global* characterization below. But as hinted, this formalization is unsatisfactorily involved. A more convenient and intelligible representation (formalization) $\mathcal{OT}$ of $\mathcal{IT}$ would be preferable. In fact, we believe such a representation to be necessary in order to get a good intuition about the 'correctness' of the formalization that could otherwise be too complex for an intelligible understanding. To reach a formalization with such properties it would be useful if each theory T_i could be characterized 'locally' towards its immediate object of study, i.e., if we could represent each 'meta/object' language relation between a theory T_i and T_{i-1} separately. We propose such a *local* characterization below and compare it with the *global* characterization.

3.3. THE FORMAL LANGUAGE

$\mathcal{OT}$ must be developed in a suitable formal language. Horovitz' unattainable perfect informal legal theory is ideal and potentially infinite. We strive to reproduce this in a formalization which likewise ultimately would be ideal and potentially infinite. We intend to formalize obtainable parts in $\mathcal{OT}$. The formal language of $\mathcal{OT}$ must be devised bearing in mind the fact that we can never achieve a perfect formalization in $\mathcal{OT}$. In particular, it must be taken into account that the formal expressions are only proposals for sentences in $\mathcal{OT}$ and that there is a precedence to the effect that sentence proposals of a certain level overrule sentence proposals of the lower adjacent level.

The provability definition in a formalization of the informal legal theory $\mathcal{IT}$ must support upward—but prevent downward—reflection. This is due to the imperfection of realistic legal systems. Both reflection principles would be allowed in Horovitz' unattainable perfect system where each T_i is a potentially infinite, enumerable and decidable set which, for each of all the meaningful rules of its language, includes either the rule itself or its negation, but not both. This assumption could be seen in the light of the *non-liquet*-prohibition stating that every legal case brought before a court has to be decided.[1] Each theory T_i expresses T_{i-1} completely in a metalanguage exactly determining its content and conforms to both reflection principles, since everything derivable from theory T_{i-1} can be 'simulated' in theory T_i and vice versa; in principle, it would be sufficient to consider only one of these theories in an ideal formalization. In reality, however, the formalization of each T_{i-1} can only be partial, imperfect and schematic and application of the rules in T_i is necessary to assess, accept or reject the rules in T_{i-1}. This corresponds to upward reflection, where something proved on an upper level is forced upon a lower level. Downward reflection is unsound, however, since something 'proved' from imperfect knowledge on a lower level cannot be forced upon an upper level, thereby perhaps contradicting rules accepted by the legal principles on this level.

In Bowen's and Kowalski's [11] amalgamation, object level provability is represented at the metalevel by a program *Pr* defining the predicate *Demo*/2. Between the levels, two reflection rules are introduced allowing an object level execution to be replaced by a metalevel execution, and vice versa. The upward reflection rule states that if there is a metaproof, then the result of this metaproof can be enforced upon the object level, and vice versa for the downward reflection rule. The two reflection rules are founded on a definition of 'representability' of provability, i.e., equivalence between object language and metalanguage proofs. This amalgamated language is inadequate for characterizing the informal legal theory $\mathcal{IT}$. Its provability predicate *Demo* is founded on representability and requires both reflection rules. *Demo represents* provability on the object level, thus presupposing that an object level proof procedure exists, which is not the case in our domain whose counterpart to *Demo defines* rather than represents a provability relation for an object language. Examples of *Demo* predicates giving *definitional* extensions of theories deviating from the representability notion are, e.g., metalevel definitions allowing upward reflection to enforce proofs on the object level which the object level theorem prover cannot carry out itself; cf. Kowalski [50]. This latter use of the *Demo* predicate accords with our understanding of the informal legal theory $\mathcal{IT}$ and could be used to characterize its hierarchical structure.

---

[1] For instance, the court may not commit *déni de justice*, i.e., reject a situation on the ground that it is not covered by any regulation; cf. Part II Ch. 13.

We have not felt any need to name open formulas, i.e., containing object variables. In fact, this has so far rendered Horn clause logic sufficient as formal language. We have realized an executable Horn clause theory $\mathcal{MT}$, whose domain consists of the meaningful expressions of the language of $\mathcal{OT}$ that in turn formalizes the informal legal theory $\mathcal{IT}$. In the formal language of $\mathcal{MT}$ we define a formalization *Demo* in $\mathcal{OT}$ of logic provability in $\mathcal{IT}$. The language of $\mathcal{OT}$ is represented by terms, i.e., terms of Horn clauses, while (meta)sentences about this language and its theories are represented by formulas, i.e., by Horn clauses. The formalization in $\mathcal{OT}$ of logic provability in $\mathcal{IT}$ is thus defined in terms of Horn clauses, and statements about this formalization are represented in Horn clauses composing the $\mathcal{MT}$ theory.

In the table below we give some examples of expressions of the language of $\mathcal{OT}$ which thus are objects of the domain of $\mathcal{MT}$ and appear as terms in $\mathcal{MT}$. The examples are from the three lowest levels of $\mathcal{OT}$.

```
level 3:
  t(3) n(t(2)) n(n(t(1))) demo(n(t(2)),n(demo(n(t(1)),n(rule1)))
                          demo(n(t(2)),n(demo(n(t(1)),n(rule2)))
                          demo(n(t(2)),
                               n(demo(n(t(1)),
                                      n(and(rule1,rule2)))))

level 2:
       t(2)    n(t(1))    demo(n(t(1)),n(rule1))
                          demo(n(t(1)),n(rule2))
                          demo(n(t(1)),n(and(rule1,rule2)))

level 1:
               t(1)       rule1
                          rule2
                          and(rule1,rule2)
```

We confine ourselves here to presenting only one formula of $\mathcal{MT}$, a statement which states that `Proof_1` is identical with a term which is a metaproof in $\mathcal{MT}$ of a formal proof in $\mathcal{OT}$.

```
Proof_1 = (sent_of(th(1),rule1):-
              proof_of(th(2),proved_that(th(1),rule1),Proof_2))
```

In the sequel, the approach mentioned above for representing $\mathcal{IT}$ is motivated by a three-step argument. Steps one and two discuss, respectively, two possible formalizations $\mathcal{OT}$ of $\mathcal{IT}$. The first is based on a *prima facie* top-down characterization of $\mathcal{IT}$ but is rather intractable; the second is a significantly more tractable transformation of the first. The third step motivates why the second formalization should in turn be defined in a separate interactive metatheory $\mathcal{MT}$.

3.4. GLOBAL VERSUS LOCAL CHARACTERIZATION

Formalization of the multilayered $\mathcal{IT}$ can be approached along several lines. It is critically important to make a realistic choice among available alternatives. Our choice is motivated in this section.

The topmost theory T_n of $\mathcal{IT}$ determines the content of all lower level theories $T_{n-1}, \ldots, T_1$. Therefore, in a direct characterization of $\mathcal{IT}$ only sentences of T_n appear. However, the interaction between theories $T_{n-1}, \ldots, T_1$ may be analysed in a metalevel theory of $\mathcal{IT}$ (or equally of T_n). A characterization of such a metalevel theory gives an indirect characterization of $\mathcal{IT}$, allowing each of $T_n, \ldots, T_1$ to appear as a separate theory. We call these two characterizations of $\mathcal{IT}$ the *global* and the *local* characterization, respectively. Since forming one exclusive theory T_n explicitly embracing the totality of $\mathcal{IT}$, sentences of the first direct characterization have a global dominance. In contrast, T_n's dominance is implicit in the second indirect characterization and sentences appear 'locally' in $T_n, \ldots, T_1$.

A globally characterized $\mathcal{OT}$ is to axiomatize the sentences of T_n and these appear as formulas in $\mathcal{OT}$. In particular, $\mathcal{OT}$ axiomatizes the sentences of T_n about logic provability for T_{n-1}. Sentences and inference rules of the lower level theories $T_{n-1}, \ldots, T_1$ appear formalized as terms in $\mathcal{OT}$. That is to say, the formulas of $\mathcal{OT}$ contain multilayered terms describing the content of these lower level theories.

Below, we first stereotype certain parts of a global characterization of $\mathcal{IT}$ into an $\mathcal{OT}$. This is followed by a derivation illustrating the complexity of such an $\mathcal{OT}$ and motivating the development of derived inference rules in a separate formal metatheory of $\mathcal{OT}$ based instead on the local characterization of $\mathcal{IT}$.

For these examples a theory T_n of some arbitrary level n has been selected as the topmost theory, thus deciding the content of all lower level theories $T_{n-1}, \ldots, T_1$. Therefore, the stereotyped fragment of $\mathcal{IT}$ has the form of a *Demo* predicate. For the sample derivation, T_4 is our choice for T_n.

Next, an $\mathcal{OT}$ based on a local characterization of $\mathcal{IT}$ is investigated. The local characterization is based on an informal metatheory of T_n analysing the interaction between its lower level theories. The rules of this informal metatheory must be identified and reproduced for a locally characterized $\mathcal{OT}$. It is proved that each SLD-refutation in a locally characterized $\mathcal{OT}$ can be expanded into a corresponding SLD-refutation in a globally characterized $\mathcal{OT}$.

$\mathcal{OT}$ based on the local characterization of $\mathcal{IT}$ can be implemented directly as a metalogic program. This is not a practical solution, however. Satisfying the need to furnish external knowledge calls for another approach. We propose a separate semiformal metatheory $\mathcal{MT}$, implemented as an interactive metalogic program. $\mathcal{MT}$ together with a user should construct an $\mathcal{OT}$ based on the local characterization of $\mathcal{IT}$.

3.4.1. The Global Characterization. In the *global* characterization, $\mathcal{IT}$

consists thus only of sentences of its topmost level. Consider a theory T_n constituting the topmost level in $\mathcal{IT}$. Since T_n describes the content of all lower level theories T_{n-1}, ..., T_1 a formalization $\mathcal{OT}$ of $\mathcal{IT}$ can be construed as a single formal theory `t(N)` representing T_n. Theory `t(N)` would contain a representation of all the formulas and terms of a formal theory `t(N-1)` representing theory T_{n-1} which in turn contains a representation of all the formulas and terms of a formal theory `t(N-2)` representing theory T_{n-2} etc., all down to the lowest theory T_1. All formulas of theories `t(N-1)`, ..., `t(1)` appear thus as terms in `t(N)`.

The first formula I below (in which $(cons)[\ldots]$ is a term level description of a universal quantifier) illustrates how we may represent schemata in a global characterization of $\mathcal{IT}$. Formula I states that all instances of the (tertiary) schema for secondary rules for *analogia legis* could be rules in theory T_2 of $\mathcal{IT}$. $\mathcal{IT}$ is globally characterized, and each instance of formula I must be a proposed sentence of T_n defined in *Demo*-relations from level $n-1$ down to level 3. Being a formula of this topmost theory, I must be externally verified as meaningful and legally acceptable and in fact as belonging to T_n.

$$
(I) \quad
\begin{array}{l}
Demo(n(T_{n-1}),\\
\quad n(Demo(n(T_{n-2}),\\
\quad\quad n(\ldots\\
\quad\quad\quad Demo(n(T_3),\\
\quad n((cons)[\\
\quad\quad (ante)[\\
\quad\quad \ldots (contra\_interests)[\\
\quad\quad\quad Demo(n(T_2),\\
\quad\quad\quad\quad n(AnalogiaLegis(n((cons \Leftarrow ante)), CommercialLaw)\\
\quad\quad\quad\quad\quad \Leftarrow and(Not(CasuisticalInterp(CommercialLaw,\\
\quad\quad\quad\quad\quad\quad\quad\quad n((cons \Leftarrow ante)))),\\
\quad\quad\quad\quad\quad\quad \ldots\\
\quad\quad\quad\quad\quad\quad and(Supports(provision\_no, \_,\\
\quad\quad\quad\quad\quad\quad\quad\quad pro\_interests, interests),\\
\quad\quad\quad\quad\quad\quad \ldots\\
\quad\quad\quad\quad\quad\quad Outweigh(pro\_interests,\\
\quad\quad\quad\quad\quad\quad\quad\quad contra\_interests)\ldots)\ldots)))\\
\quad\quad\quad]\ldots]]))))\ldots))))
\end{array}
$$

In the global characterization T_n must have means for properly treating universal quantification at the respective levels. Formula I illustrates the inappropriateness of the implicit prefix universal quantifiers of Horn clause logic. Moreover, since *cons*, *ante*, ..., *contra_interests* are schema variables they are quantified at the tertiary level (in theory T_3) and not at the secondary level which at a first glance perhaps would have seemed more natural. Apart from rule schemata and rules of logic, $\mathcal{IT}$ is assumed to contain ground rules only. A schema for rules of a theory T_i appears at the metalevel with respect to T_i. Its variables range over the names for expressions of the language of T_i, but not over the individuals in the domain of T_i. Schema variables of theory T_i are thus individual variables of theory T_{i+1}. (It is worth noting that this reading of the quantifiers has a clear affinity to the substitutional, as contrasted to the objectual interpretation of quantifiers; cf. e.g., [28].)

An even more illustrative example of global characterization is the representation in T_n of a rule of inference. T_n consists only of formulas of its own level and must therefore have one particular formula encoding each inference rule for each of the theories $T_1, \ldots, T_{n-1}$. For instance, in T_n and-introduction is defined in *Demo*-relations from level $n-1$ down to level 2, i.e.,

$$\begin{aligned}&\forall \mathrm{a} \forall \mathrm{b}[\mathit{Demo}(n(T_{n-1}), n(\mathit{and}(\mathrm{a},\mathrm{b})))\\&\qquad \leftarrow \mathit{Demo}(n(T_{n-1}), n(\mathrm{a})) \wedge \mathit{Demo}(n(T_{n-1}), n(\mathrm{b}))]\end{aligned}$$

and

$$(II)\qquad\begin{aligned}&\mathit{Demo}(n(T_{n-1}),\\&\quad n((a)[(b)[\mathit{Demo}(n(T_{n-2}), n(\mathit{and}(a,b)))\\&\qquad\qquad \Leftarrow \mathit{and}(\mathit{Demo}(n(T_{n-2}), n(a)),\\&\qquad\qquad\qquad \mathit{Demo}(n(T_{n-2}), n(b)))]]))\end{aligned}$$

and ...

$$\begin{aligned}&\mathit{Demo}(n(T_{n-1}),\\&\quad n(\mathit{Demo}(n(T_{n-2}),\\&\qquad n(\ldots\\&\qquad\quad \mathit{Demo}(n(T_2),\\&\qquad\qquad n((a)[(b)[\mathit{Demo}(n(T_1), n(\mathit{and}(a,b)))\\&\qquad\qquad\qquad \Leftarrow \mathit{and}(\mathit{Demo}(n(T_1), n(a)),\\&\qquad\qquad\qquad\quad \mathit{Demo}(n(T_1), n(b)))]])) \ldots))))\end{aligned}$$

Note the difference between on the one hand $\forall$a, $\leftarrow$, and $\wedge$ and on the other (a), $\Leftarrow$, and $\mathit{and}(\ldots,\ldots)$. The former are logic symbols for, in turn, universal quantification, implication and conjunction, the latter are term level descriptions of such logic symbols only. Thus, the theorem prover carrying out the proofs in T_n interprets the former as logic constructs while the latter are meaningless to it.

The complexity this causes can be illustrated by a derivation in the global characterization of $\mathcal{IT}$ involving reflection between adjacent levels. First we briefly sketch the planned steps in the derivation. Suppose we have a query

$$
\begin{aligned}
&Demo(n(T_{n-1}),\\
&\quad n(Demo(n(T_{n-2}),\\
&\qquad n(\ldots\\
&\qquad\quad Demo(n(T_2),\\
&\qquad\qquad n(Demo(n(T_1),\\
&\qquad\qquad\quad n((Cons(A,B) \Leftarrow Ante(A,B)))))) \ldots))))
\end{aligned}
$$

where $Cons(A,B) \Leftarrow Ante(A,B)$ is a ground rule being proposed as a potential primary rule.

Whether this rule proposal is acceptable or not is determined by (proposals for) secondary rules, so upward reflection is required. The global characterization does not require any explicit reflection rule. Instead, reflection is realized through interaction between inference rules at separate levels. In this case the particular *modus ponens* inference rule for level 2, i.e.,

$$
\begin{aligned}
&Demo(n(T_{n-1}),\\
&\quad n(Demo(n(T_{n-2}),\\
&\qquad n(\ldots\\
&\qquad\quad Demo(n(T_3),\\
&\qquad\qquad n((a)[(b)[Demo(n(T_2), n(a))\\
&\qquad\qquad\quad \Leftarrow and(Demo(n(T_2), n((a \Leftarrow b))),\\
&\qquad\qquad\qquad Demo(n(T_2), n(b)))]])) \ldots))))
\end{aligned}
$$

will at some point in the derivation mediate that resolution occurs at some later point with

$$
\begin{aligned}
&Demo(n(T_{n-1}),\\
&\quad n(Demo(n(T_{n-2}),\\
&\qquad n(\ldots\\
&\qquad\quad Demo(n(T_2),\\
&\qquad\qquad n((cons)[(ante)[Demo(n(T_1), n((cons \Leftarrow ante)))\\
&\qquad\qquad\quad \Leftarrow AnalogiaLegis(n((cons \Leftarrow ante)))\\
&\qquad\qquad\quad]])) \ldots))))
\end{aligned}
$$

which is a (tertiary schema for) a secondary rule stating that an expression can be assumed as a sentence of T_1 if this accords with *analogia legis*. Further application of *modus ponens* for level 2 yields

$$
\begin{aligned}
&\mathit{Demo}(n(T_{n-1}),\\
&\qquad n(\mathit{Demo}(n(T_{n-2}),\\
&\qquad\qquad n(\ldots\\
&\qquad\qquad\quad \mathit{Demo}(n(T_2),\\
&\qquad\qquad\qquad n(\mathit{AnalogiaLegis}(n((\mathit{Cons} \Leftarrow \mathit{Ante})))))\ldots))))
\end{aligned}
$$

which at some step in the computation should resolve with formula I, etc.

We now reconstruct the steps in this derivation. Let T_n be T_4 and for typographical reasons let D denote *Demo*.

With this notation inference rules involved are

MP$_4$: $\forall \mathrm{a}[\forall \mathrm{b}[D(n(T_4), n(\mathrm{a})) \leftarrow D(n(T_4), n((\mathrm{a} \Leftarrow \mathrm{b}))) \wedge D(n(T_4), n(\mathrm{b}))]]$
MP$_3$: $D(n(T_4), n((a)[(b)[$
$\quad (D(n(T_3), n(a)) \Leftarrow and(D(n(T_3), n((a \Leftarrow b))), D(n(T_3), n(b))))]]))$
MP$_2$: $D(n(T_4), n(D(n(T_3), n((a)[(b)[$
$\quad (D(n(T_2), n(a)) \Leftarrow and(D(n(T_2), n((a \Leftarrow b))), D(n(T_2), n(b))))]]))))$

ANDI$_4$: $\forall \mathrm{a}[\forall \mathrm{b}[D(n(T_4), n(and(\mathrm{a}, \mathrm{b}))) \leftarrow D(n(T_4), n(\mathrm{a})) \wedge D(n(T_4), n(\mathrm{b}))]]$
ANDI$_3$: $D(n(T_4), n((a)[(b)[$
$\quad (D(n(T_3), n(and(a, b))) \Leftarrow and(D(n(T_3), n(a)), D(n(T_3), n(b))))]]))$
ANDI$_2$: $D(n(T_4), n(D(n(T_3), n((a)[(b)[$
$\quad (D(n(T_2), n(and(a, b))) \Leftarrow and(D(n(T_2), n(a)), D(n(T_2), n(b))))]]))))$

UE$_4$: $\forall \mathrm{q}[\forall \mathrm{p}[\forall \mathrm{x}[\forall \mathrm{y}[$
$\quad D(n(T_4), n(\mathrm{q})) \leftarrow D(n(T_4), n((\mathrm{x})[\mathrm{p}])) \wedge Subst(\mathrm{x}, \mathrm{p}, \mathrm{y}, \mathrm{q})]]]]$
UE$_3$: $D(n(T_4), n((q)[(p)[(x)[(y)[$
$\quad (D(n(T_3), n(q)) \Leftarrow and(D(n(T_3), n((x)[p])), Subst(x, p, y, q)))]]]]))$
UE$_2$: $D(n(T_4), n(D(n(T_3), n((q)[(p)[(x)[(y)[$
$\quad (D(n(T_2), n(q)) \Leftarrow and(D(n(T_2), n((x)[p])), Subst(x, p, y, q)))]]]]))))$

Readability is enhanced by letting linguistic expressions name themselves autonymously. This gives the inference rules of Table 3.1, in which we have generalized the indices for an arbitrary topmost theory T_n.

We first show the derivation excluding the group of inference rules for universal elimination. Afterwards, the first segment of the derivation will be expanded to illustrate how universal elimination is coped with and the additional complexity this gives rise to.

$$
\begin{array}{ll}
\mathrm{MP}_n: & \forall \mathrm{a}[\forall \mathrm{b}[D(T_{n-1}, \mathrm{a}) \leftarrow D(T_{n-1}, (\mathrm{a} \Leftarrow \mathrm{b})) \wedge D(T_{n-1}, \mathrm{b})]] \\
\mathrm{MP}_{n-1}: & D(T_{n-1}, (a)[(b)[\\
& \quad (D(T_{n-2}, a) \Leftarrow \mathit{and}(D(T_{n-2}, (a \Leftarrow b)), D(T_{n-2}, b)))]]) \\
\mathrm{MP}_{n-2}: & D(T_{n-1}, D(T_{n-2}, (a)[(b)[\\
& \quad (D(T_{n-3}, a) \Leftarrow \mathit{and}(D(T_{n-3}, (a \Leftarrow b)), D(T_{n-3}, b)))]])) \\
\vdots & \\
\mathrm{MP}_{n-i}: & D(T_{n-1}, \ldots D(T_{n-i}, (a)[(b)[\\
& \quad (D(T_{n-(i+1)}, a) \Leftarrow \mathit{and}(D(T_{n-(i+1)}, (a \Leftarrow b)), D(T_{n-(i+1)}, b)))]]) \ldots) \\
\vdots & \\
\mathrm{ANDI}_n: & \forall \mathrm{a}[\forall \mathrm{b}[D(T_{n-1}, \mathit{and}(\mathrm{a}, \mathrm{b})) \leftarrow D(T_{n-1}, \mathrm{a}) \wedge D(T_{n-1}, \mathrm{b})]] \\
\mathrm{ANDI}_{n-1}: & D(T_{n-1}, (a)[(b)[\\
& \quad (D(T_{n-2}, \mathit{and}(a, b)) \Leftarrow \mathit{and}(D(T_{n-2}, a), D(T_{n-2}, b)))]]) \\
\mathrm{ANDI}_{n-2}: & D(T_{n-1}, D(T_{n-2}, (a)[(b)[\\
& \quad (D(T_{n-3}, \mathit{and}(a, b)) \Leftarrow \mathit{and}(D(T_{n-3}, a), D(T_{n-3}, b)))]])) \\
\vdots & \\
\mathrm{ANDI}_{n-i}: & D(T_{n-1}, \ldots D(T_{n-i}, (a)[(b)[\\
& \quad (D(T_{n-(i+1)}, \mathit{and}(a, b)) \Leftarrow \mathit{and}(D(T_{n-(i+1)}, a), \\
& \qquad\qquad D(T_{n-(i+1)}, b)))]]) \ldots) \\
\vdots & \\
\mathrm{UE}_n: & \forall \mathrm{q}[\forall \mathrm{p}[\forall \mathrm{x}[\forall \mathrm{y}[D(T_{n-1}, \mathrm{q}) \leftarrow D(T_{n-1}, (\mathrm{x})[\mathrm{p}]) \wedge \mathit{Subst}(\mathrm{x}, \mathrm{p}, \mathrm{y}, \mathrm{q})]]]] \\
\mathrm{UE}_{n-1}: & D(T_{n-1}, (q)[(p)[(x)[(y)[\\
& \quad (D(T_{n-2}, q) \Leftarrow \mathit{and}(D(T_{n-2}, (x)[p]), \mathit{Subst}(x, p, y, q)))]]]]) \\
\mathrm{UE}_{n-2}: & D(T_{n-1}, D(T_{n-2}, (q)[(p)[(x)[(y)[\\
& \quad (D(T_{n-3}, q) \Leftarrow \mathit{and}(D(T_{n-3}, (x)[p]), \mathit{Subst}(x, p, y, q)))]]]])) \\
\vdots & \\
\mathrm{UE}_{n-i}: & D(T_{n-1}, \ldots D(T_{n-i}, (q)[(p)[(x)[(y)[\\
& \quad (D(T_{n-(i+1)}, q) \Leftarrow \mathit{and}(D(T_{n-(i+1)}, (x)[p]), \mathit{Subst}(x, p, y, q)))]]]]) \ldots) \\
\vdots &
\end{array}
$$

Table 3.1. Inference rules and term level descriptions of such rules.

The first segment is

$$
\begin{array}{ll}
\leftarrow \underline{D(T_4, D(T_3, D(T_2, D(T_1, \mathit{rule}))))} & \\
\leftarrow \underline{D(T_4, (D(T_3, D(T_2, D(T_1, \mathit{rule}))) \Leftarrow \mathrm{b}))} \wedge D(T_4, \mathrm{b}) & \mathrm{MP}_4 \\
\quad \{\mathrm{b} \mapsto \mathit{and}(D(T_3, (D(T_2, D(T_1, \mathit{rule})) \Leftarrow b')), D(T_3, b'))\} & \\
\leftarrow D(T_4, \mathit{and}(D(T_3, (D(T_2, D(T_1, \mathit{rule})) \Leftarrow b')), D(T_3, b'))) & \mathrm{MP}_3
\end{array}
$$

where *rule* corresponds to $(Cons(A, B) \Leftarrow Ante(A, B))$. After this segment the subgoal $D(T_4, \mathrm{b})$ is instantiated to $D(T_4, and(D(T_3, (D(T_2, D(T_1, rule)) \Leftarrow b')), D(T_3, b')))$. Note the difference between b and b'. Whereas b is interpreted as a logic variable, b' is only an arbitrary term to the theorem prover processing the formalization of T_n. In T_n, UE_4 gives an axiomatization of the interpretation of b' as a variable by the *Subst*-predicate. How this axiomatization is processed by the theorem prover is explained when this segment is expanded with universal elimination.

The further computation appears thus

$$\begin{array}{ll}
\leftarrow \underline{D(T_4, and(D(T_3, (D(T_2, D(T_1, rule)) \Leftarrow b')), D(T_3, b')))} & \\
\leftarrow \underline{D(T_4, D(T_3, (D(T_2, D(T_1, rule)) \Leftarrow b')))} \wedge D(T_4, D(T_3, b')) & \mathrm{ANDI}_4 \\
\quad \{b' \mapsto and(D(T_2, (D(T_1, rule) \Leftarrow b'')), D(T_2, b''))\} & \\
\leftarrow D(T_4, D(T_3, and(D(T_2, (D(T_1, rule) \Leftarrow b'')), D(T_2, b'')))) & \mathrm{MP}_2
\end{array}$$

The subgoal $D(T_4, D(T_3, b'))$ has after this segment become instantiated to $D(T_4, D(T_3, and(D(T_2, (D(T_1, rule) \Leftarrow b'')), D(T_2, b''))))$ whose further computation is rather tricky. Note that the current goal does not unify ANDI_3. Resolution with MP_4 must occur first to obtain two subgoals. The first unifies ANDI_3 and as a result the second will contain the instantiated 'antecedent' of the 'implication' in ANDI_3. This 'implication' is only an arbitrary term for the theorem prover.

$$\begin{array}{ll}
\leftarrow \underline{D(T_4, D(T_3, and(D(T_2, (D(T_1, rule) \Leftarrow b'')), D(T_2, b''))))} & \\
\leftarrow \underline{D(T_4, (D(T_3, and(D(T_2, (D(T_1, rule) \Leftarrow b'')), D(T_2, b''))) \Leftarrow \mathrm{b}'''))} & \mathrm{MP}_4 \\
\quad \wedge D(T_4, \mathrm{b}''') & \\
\qquad \{\mathrm{b}''' \mapsto and(D(T_3, D(T_2, (D(T_1, rule) \Leftarrow b''))), D(T_3, D(T_2, b'')))\} & \\
\leftarrow \underline{D(T_4, and(D(T_3, D(T_2, (D(T_1, rule) \Leftarrow b''))), D(T_3, D(T_2, b''))))} & \mathrm{ANDI}_3 \\
\leftarrow \underline{D(T_4, D(T_3, D(T_2, (D(T_1, rule) \Leftarrow b''))))} \wedge D(T_4, D(T_3, D(T_2, b''))) & \mathrm{ANDI}_4 \\
\qquad \{b'' \mapsto AnalogiaLegis(rule)\} & \\
\leftarrow D(T_4, D(T_3, D(T_2, AnalogiaLegis(rule)))) & \text{AL fact}
\end{array}$$

Now we have reached what we set out to do. This illustrates the intractability of the ground term level representation imposed by the global characterization. Moreover, we have excluded the really awkward element of universal elimination. We confine ourselves to expanding the first segment above to illustrate this problem. It becomes

$$\begin{array}{ll}
\leftarrow \underline{D(T_4, D(T_3, D(T_2, D(T_1, rule))))} & \\
\leftarrow \underline{D(T_4, (D(T_3, D(T_2, D(T_1, rule))) \Leftarrow \mathrm{b}))} \wedge D(T_4, \mathrm{b}) & \mathrm{MP}_4 \\
\leftarrow \underline{D(T_4, (\mathrm{x})[\mathrm{p}])} & \mathrm{UE}_4
\end{array}$$

$$
\begin{aligned}
&\quad \wedge\, \mathit{Subst}(\mathrm{x},\mathrm{p},\mathrm{b}',(D(T_3,D(T_2,D(T_1,\mathit{rule})))\Leftarrow \mathrm{b}))\wedge D(T_4,\mathrm{b}) \\
&\leftarrow \underline{D(T_4,(\mathrm{x}')[\mathrm{p}'])}\wedge \mathit{Subst}(\mathrm{x}',\mathrm{p}',\mathrm{a}',(\mathrm{x})[\mathrm{p}]) && \mathrm{UE}_4\\
&\quad \wedge\, \mathit{Subst}(\mathrm{x},\mathrm{p},\mathrm{b}',(D(T_3,D(T_2,D(T_1,\mathit{rule})))\Leftarrow \mathrm{b}))\wedge D(T_4,\mathrm{b}) \\
&\quad\ \{\mathrm{x}'\mapsto a,\mathrm{p}'\mapsto (b)[(D(T_3,a)\Leftarrow \mathit{and}(D(T_3,(a\Leftarrow b)),D(T_3,b)))]\}\\
&\leftarrow \underline{\mathit{Subst}(a,(b)[(D(T_3,a)\Leftarrow \mathit{and}(D(T_3,(a\Leftarrow b)),D(T_3,b)))],\mathrm{a}',(\mathrm{x})[\mathrm{p}])} && \mathrm{MP}_3\\
&\quad \wedge\, \mathit{Subst}(\mathrm{x},\mathrm{p},\mathrm{b}',(D(T_3,D(T_2,D(T_1,\mathit{rule})))\Leftarrow \mathrm{b}))\wedge D(T_4,\mathrm{b}) \\
&\quad\ \{\mathrm{x}\mapsto b,\mathrm{p}\mapsto (D(T_3,\mathrm{a}')\Leftarrow \mathit{and}(D(T_3,(\mathrm{a}'\Leftarrow b)),D(T_3,b)))\}\\
&\leftarrow \underline{\mathit{Subst}(b,(D(T_3,\mathrm{a}')\Leftarrow \mathit{and}(D(T_3,(\mathrm{a}'\Leftarrow b)),D(T_3,b))),}\\
&\qquad\quad \underline{\mathrm{b}',(D(T_3,D(T_2,D(T_1,\mathit{rule})))\Leftarrow \mathrm{b}))}\wedge D(T_4,\mathrm{b}) && \mathrm{Subst}\\
&\quad\ \left\{\begin{array}{l}\mathrm{a}'\mapsto D(T_2,D(T_1,\mathit{rule})),\\ \mathrm{b}\mapsto \mathit{and}(D(T_3,(D(T_2,D(T_1,\mathit{rule}))\Leftarrow b')),D(T_3,b'))\\ \mathrm{b}'\mapsto b'\end{array}\right\}\\
&\leftarrow D(T_4,\mathit{and}(D(T_3,(D(T_2,D(T_1,\mathit{rule}))\Leftarrow b')),D(T_3,b'))) && \mathrm{Subst}
\end{aligned}
$$

After this segment the subgoal $D(T_4, \mathrm{b})$ has become instantiated to

$$D(T_4,\mathit{and}(D(T_3,(D(T_2,D(T_1,\mathit{rule}))\Leftarrow b')),D(T_3,b')))$$

the further computation of which corresponds to the segments above expanded by universal elimination.

In the global characterization, the hierarchical structure of the informal legal theory $\mathcal{IT}$ causes a significant complexity in its formalization $\mathcal{OT}$. This complexity would be reduced, however, if instead it were possible to formalize each T_i of $\mathcal{IT}$ as a separate theory. That is to say, each T_i is characterized '*locally*' towards its immediate object of study, i.e., each 'meta/object' language relation in $\mathcal{IT}$ is represented separately.

3.4.2. The Local Characterization. In the local characterization a link between adjacent levels is obtained by characterizing the *Demo* predicate as defining the provability relation on a lower level. For example, the formula

$$\mathit{Demo}(n(T_1), n((J \Leftarrow C)))$$

says (on level 2) that the primary rule $J \Leftarrow C$ is provable from, and thus included in, the object theory T_1. Stated as a goal the formula reads 'is $J \Leftarrow C$ provable from the theory T_1?' the proof of which corresponds to a line of arguments to the effect that the inclusion in T_1 of $J \Leftarrow C$ should be regarded as in accordance with the (secondary) metarules of legal interpretation, thus showing, as hinted on p. 14, that the rule is legally acceptable. The provability definition

$$
\begin{aligned}
&\mathit{Demo}(n(T_i), n(A))\\
&\leftarrow \mathit{Demo}(n(T_{i+1}), n(\mathit{Demo}(n(T_i), n(A))))
\end{aligned}
$$

expresses the conditions for including $J \Leftarrow C$ in T_1, i.e., the secondary rules whose inclusion in T_2, in turn, depends on theories of higher levels, whose provability definitions are characterized in a similar way yielding a whole hierarchy of interdepending provability definitions, still however allowing us to describe and consider each T_i as a separate theory.

The declarative reading of the formula $Demo(n(T_1), n((J \Leftarrow C)))$ is rendered by taking T_1 as a static theory implicitly consisting of all rules fulfilling the conditions for inclusion. Only the boundary between rules shown to satisfy the conditions and those not yet shown to satisfy the conditions moves just as in Sergot's 'query the user' [66] the boundary between the facts given to the computer by the user and those not yet given, moves. With this reading the provability relation $T_i \vdash A_i$ coincides with the rules of logic on each level i and, though definitional, so does its formalization *Demo*.

Let us now compare the above globally characterized formulas with their counterparts in the local characterization of $\mathcal{IT}$. These counterparts are particular theorems of T_n, namely those obtained by reflecting down the *Demo*-relation of theory T_n to a level where the formulas may be implemented as pure Prolog [15] programs relying on Prolog unification for ∀-instantiation and on the usual query-mechanism of Prolog for coping with queries about proposed primary rules. So the T_n clause I above will have as its counterpart the clause

$$
(I') \quad
\begin{array}{l}
\forall \text{cons}[\forall \text{ante}[\ldots \forall \text{contra\_interests}[\\
\quad Demo(n(T_2),\\
\qquad n(AnalogiaLegis(n((\text{cons} \Leftarrow \text{ante})), CommercialLaw)\\
\qquad\quad \Leftarrow and(Not(CasuisticalInterp(CommercialLaw,\\
\qquad\qquad\qquad\qquad n((\text{cons} \Leftarrow \text{ante})))),\\
\qquad\qquad \ldots\\
\qquad\qquad and(Supports(\text{provision\_no}, \_, \text{pro\_interests}, \text{interests}),\\
\qquad\qquad \ldots\\
\qquad\qquad Outweigh(\text{pro\_interests}, \text{contra\_interests})\ldots)\ldots)))\\
\quad]\ldots]]
\end{array}
$$

on level 3, i.e., the clause is reflected down to theory T_3, where 'cons', 'ante',..., 'contra_interests' may appear as logic variables since Prolog copes with the prefix ∀ quantifier. Note that the definition of the provability relation between adjacent levels implies that this clause is still dependent on upper levels. Observe, that in a model for the clause, the metavariables range over the domain of formulas of theory T_2 so no confusion arises between meta and object variables.

The complexity of the global characterization of T_n is contrasted even better with that of the local characterization by comparing their respective representation of inference rules. The *Demo*-clauses *II* above are transformed into one general clause

$$(II') \qquad \begin{aligned} &\forall \mathrm{t} \forall \mathrm{a} \forall \mathrm{b}[\mathit{Demo}(n(\mathrm{t}), n(\mathit{and}(\mathrm{a}, \mathrm{b}))) \\ &\quad \leftarrow \mathit{Demo}(n(\mathrm{t}), n(\mathrm{a})) \\ &\quad \wedge \mathit{Demo}(n(\mathrm{t}), n(\mathrm{b})). \end{aligned}$$

A word of caution is in order about the local characterization of $\mathcal{IT}$. The correctness of the transformation from T_n to locally characterized clauses presupposes that each such clause is taken as a separate first-order theory. For example, the intended interpretation for a clause $\forall a[\mathit{Demo}(n(T_2), n(a))]$ cannot be a model for a clause $\forall b[\mathit{Demo}(n(T_3), n(b))]$ since a and b range over different domains. Including such clauses in the same theory (program) necessitates amplifications, such as the typed representation proposed by Hill and Lloyd [42]. For example, clause I' could be represented as

$$\begin{aligned}
&\forall \mathrm{cons}[\forall \mathrm{ante}[\ldots \forall \mathrm{contra_interests}[\\
&\quad \mathit{Demo}(n(T_2),\\
&\qquad n(\mathit{AnalogiaLegis}(n((\mathrm{cons} \Leftarrow \mathrm{ante})), \mathit{CommercialLaw})\\
&\qquad\quad \Leftarrow \mathit{and}(\mathit{Not}(\mathit{CasuisticalInterp}(\mathit{CommercialLaw},\\
&\qquad\qquad\qquad n((\mathrm{cons} \Leftarrow \mathrm{ante})))),\\
&\qquad\qquad \ldots\\
&\qquad\qquad \mathit{and}(\mathit{Supports}(\mathrm{provision_no}, _, \mathrm{pro_interests}, \mathrm{interests}),\\
&\qquad\qquad \ldots\\
&\qquad\qquad \mathit{Outweigh}(\mathrm{pro_interests}, \mathrm{contra_interests}) \ldots) \ldots)))\\
&\quad \leftarrow \mathit{RangeOverExpressionsOf}(n(T_2), [\mathrm{cons}, \ldots, \mathrm{contra_interests}])\\
&\quad] \ldots]]
\end{aligned}$$

where *RangeOverExpressionsOf* is a predicate holding for all individuals in the domain of T_3, i.e., all expressions of T_2.

The following theorem proves that the local characterization is a correct metatheory of the global characterization.

Call the local and global characterization of $\mathcal{OT}$, respectively, $\mathcal{LOT}$ and $\mathcal{GOT}$.

Theorem 1. *Each SLD-refutation in $\mathcal{LOT}$ of a goal*

$$\leftarrow D(T_{n-i}, \phi_{n-i})$$

can be expanded into a corresponding SLD-refutation in $\mathcal{GOT}$ of a goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots).$$

Proof. In $\mathcal{LOT}$ only three conditional clauses exist: MP, ANDI and UP, each of which encodes one inference rule. In $\mathcal{GOT}$, conditional clauses exist for MP_n, ANDI_n and UE_n, and unit clauses for MP_{n-i}, ANDI_{n-i}, and UE_{n-i} for all i, $1 \leq i < n-1$. Suppose $\mathcal{GOT}$ is axiomatized in $\mathcal{LOT}$. Each non-logical axiom

$$D(T_{n-i}, \Phi_{n-i}) \leftarrow$$

in $\mathcal{LOT}$ is then expressed in the theory T_n of $\mathcal{GOT}$ as the corresponding non-logical axiom

$$D(T_{n-1}, \ldots D(T_{n-i}, \Phi_{n-i}) \ldots) \leftarrow .$$

Moreover, MP and ANDI in $\mathcal{LOT}$ will be the MP_n and ANDI_n for T_n in $\mathcal{GOT}$.

Consider an SLD-refutation , $^{\mathcal{LOT}}$ of an arbitrary goal

$$\leftarrow D(T_{n-i}, \phi_{n-i})$$

in $\mathcal{LOT}$. , $^{\mathcal{LOT}}$ may succeed by resolution with a unit clause

$$D(T_{n-i}, \phi_{n-i}) \leftarrow, \qquad \text{(Case 0)}$$

or with the conditional clause MP

$$D(T_{n-i}, \phi_{n-i}) \leftarrow D(T_{n-i}, (\phi_{n-i} \Leftarrow \psi_{n-i})) \wedge D(T_{n-i}, \psi_{n-i}), \qquad \text{(Case 1)}$$

or, if ϕ_{n-i} has the form $and(\alpha_{n-i}, \beta_{n-i})$, with the conditional clause ANDI

$$D(T_{n-i}, and(\alpha_{n-i}, \beta_{n-i})) \leftarrow D(T_{n-i}, \alpha_{n-i}) \wedge D(T_{n-i}, \beta_{n-i}), \qquad \text{(Case 2)}$$

or with the conditional clause UP

$$D(T_{n-i}, \phi_{n-i}) \leftarrow D(T_{n-(i-1)}, D(T_{n-i}, \phi_{n-i})), \qquad \text{(Case 3)}$$

provided that, for each of cases 1, 2, and 3, an SLD-refutation exists for their respective bodies, i.e., any of cases 0, 1, 2, or 3 applies recursively to the goals in the body.

We must prove that , $^{\mathcal{LOT}}$ can be expanded into an SLD-refutation , $^{\mathcal{GOT}}$ in $\mathcal{GOT}$ of a goal $\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots)$ corresponding to the arbitrary goal $\leftarrow D(T_{n-i}, \phi_{n-i})$ in , $^{\mathcal{LOT}}$. Each resolution step in , $^{\mathcal{LOT}}$ has corresponding resolution steps in , $^{\mathcal{GOT}}$.

For case 0, , $^{\mathcal{GOT}}$ will succeed by resolution with a unit clause

$$D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots) \leftarrow$$

corresponding to the clause $D(T_{n-i}, \phi_{n-i}) \leftarrow$ in , $^{\mathcal{LOT}}$.

For case 1, , $^{\mathcal{GOT}}$ will include resolution steps leading to the composite goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, (\phi_{n-i} \Leftarrow \psi_{n-i})) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-i}, \psi_{n-i}) \ldots),$$

which corresponds to the composite goal

$$\leftarrow D(T_{n-i}, (\phi_{n-i} \Leftarrow \psi_{n-i})) \wedge D(T_{n-i}, \psi_{n-i})$$

in , $^{\mathcal{LOT}}$.

For case 2, where ϕ_{n-i} is of the form $and(\alpha_{n-i}, \beta_{n-i})$, , $^{\mathcal{GOT}}$ will include resolution steps leading to the composite goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, \alpha_{n-i}) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-i}, \beta_{n-i}) \ldots),$$

which corresponds to the composite goal

$$\leftarrow D(T_{n-i}, \alpha_{n-i}) \wedge D(T_{n-i}, \beta_{n-i})$$

in , $^{\mathcal{LOT}}$.

For case 3, , $^{\mathcal{GOT}}$ will include resolution steps leading to the goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(i-1)}, D(T_{n-i}, \phi_{n-i})) \ldots)$$

corresponding to the goal

$$\leftarrow D(T_{n-(i-1)}, D(T_{n-i}, \phi_{n-i}))$$

in , $^{\mathcal{LOT}}$.

Cases 0 and 3 are obvious, the latter since the goals are tautologous. Cases 1 and 2 are proved in lemmata 1 and 2.

These are all cases. Hence

if $\mathcal{LOT} \vdash D(T_{n-i}, \phi_{n-i})$
then $\mathcal{GOT} \vdash D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots)$. ∎

Lemma 1. *Every SLD-resolution step*

$$\leftarrow D(T_{n-i}, \phi_{n-i})$$
$$\leftarrow D(T_{n-i}, (\phi_{n-i} \Leftarrow \psi_{n-i})) \wedge D(T_{n-i}, \psi_{n-i}) \quad \text{(by Case 1)}$$

of , $^{\mathcal{LOT}}$ *can be constructed in* , $^{\mathcal{GOT}}$ *by a corresponding sequence of steps*

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots)$$
$$\vdots$$
$$\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, (\phi_{n-i} \Leftarrow \psi_{n-i})) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-i}, \psi_{n-i}) \ldots).$$

Proof. The proof will be by induction on the number of segments of $,^{\mathcal{GOT}}$. In order to make discernible the induction hypothesis and the induction step we need to give an illustration of how $,^{\mathcal{GOT}}$ is construed for $1 \leq i \leq 6$.

In $,^{\mathcal{GOT}}$ let

$$\leftarrow D(T_{n-1}, \gamma) \text{ be } \leftarrow D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots)$$

and

$$\begin{array}{c} \leftarrow D(T_{n-1}, \alpha) \\ \wedge D(T_{n-1}, \beta) \end{array} \text{ be } \begin{array}{l} \leftarrow D(T_{n-1}, \ldots D(T_{n-i}, (\phi_{n-i} \Leftarrow \psi_{n-i})) \ldots) \\ \wedge D(T_{n-1}, \ldots D(T_{n-i}, \psi_{n-i}) \ldots). \end{array}$$

Consider resolution steps in $,^{\mathcal{GOT}}$ between an arbitrary goal $\leftarrow D(T_{n-1}, \gamma)$ and the composite goal $\leftarrow D(T_{n-1}, \alpha) \wedge D(T_{n-1}, \beta)$, for $\gamma =$

$n-1$	ϕ_{n-1},
$n-2$	$D(T_{n-2}, \phi_{n-2})$,
$n-3$	$D(T_{n-2}, D(T_{n-3}, \phi_{n-3}))$,
$n-4$	$D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, \phi_{n-4})))$,
$n-5$	$D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, D(T_{n-5}, \phi_{n-5}))))$, and
$n-6$	$D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, D(T_{n-5}, D(T_{n-6}, \phi_{n-6})))))$

and correspondingly for α and β.

Henceforth, the primes $'$ on MP'_{n-k}, ANDI'_{n-k}, and UE'_{n-k} denote that these are goals from which all universal quantifiers or term level descriptions of such are eliminated. It is proved below that these goals are refutable from clauses MP_{n-k}, ANDI_{n-k} and UE_{n-k}, and therefore we occasionally refer to them as clauses instead of goals.

$,^{\mathcal{GOT}}$ is expanded for segments $n-1$ to $n-6$ below. Each MP'_{n-k} and ANDI'_{n-k} $(n-1 > k \geq 0)$ stands for a resolution step between one of these 'clauses' and the underlined goal. For each $n-1$ to $n-4$, a numbered bar indicates the point at which resolution would occur with its particular $\leftarrow D(T_{n-1}, \alpha) \wedge D(T_{n-1}, \beta)$. In the initial query $\leftarrow D(T_{n-1}, \phi_{n-1})$, the ϕ_{n-1} is replaced by $D(T_{n-2}, \phi_{n-2})$ for case $n-2$, by $D(T_{n-2}, D(T_{n-3}, \phi_{n-3}))$ for case $n-3$, etc.

$$\leftarrow \underline{D(T_{n-1}, \phi_{n-1})}$$

$$\leftarrow \underline{D(T_{n-1}, (\phi_{n-1} \Leftarrow \psi_{n-1}))} \wedge D(T_{n-1}, \psi_{n-1}) \qquad \mathrm{MP}'_{n}$$

——————— $n-1$ ———————

$$\leftarrow \underline{D(T_{n-1}, and(D(T_{n-2}, (\phi_{n-2} \Leftarrow \psi_{n-2})), D(T_{n-2}, \psi_{n-2})))} \qquad \mathrm{MP}'_{n-1}$$

$$\leftarrow \underline{D(T_{n-1}, D(T_{n-2}, (\phi_{n-2} \Leftarrow \psi_{n-2})))} \wedge D(T_{n-1}, D(T_{n-2}, \psi_{n-2})) \qquad \mathrm{ANDI}'_{n}$$

——————— $n-2$ ———————

$$\leftarrow \underline{D(T_{n-1}, D(T_{n-2}, \mathit{and}(D(T_{n-3}, (\phi_{n-3} \Leftarrow \psi_{n-3})), D(T_{n-3}, \psi_{n-3}))))} \quad \mathrm{MP}'_{n-2}$$

$$\leftarrow \underline{D(T_{n-1}, (D(T_{n-2}, \mathit{and}(D(T_{n-3}, (\phi_{n-3} \Leftarrow \psi_{n-3})),} \quad \mathrm{MP}'_{n}$$
$$\underline{D(T_{n-3}, \psi_{n-3}))) \Leftarrow \xi))} \wedge D(T_{n-1}, \xi)$$

$$\leftarrow \underline{D(T_{n-1}, \mathit{and}(D(T_{n-2}, D(T_{n-3}, (\phi_{n-3} \Leftarrow \psi_{n-3}))),} \quad \mathrm{ANDI}'_{n-1}$$
$$\underline{D(T_{n-2}, D(T_{n-3}, \psi_{n-3}))))}$$

$$\leftarrow \underline{D(T_{n-1}, D(T_{n-2}, D(T_{n-3}, (\phi_{n-3} \Leftarrow \psi_{n-3}))))} \quad \mathrm{ANDI}'_{n}$$
$$\wedge D(T_{n-1}, D(T_{n-2}, D(T_{n-3}, \psi_{n-3})))$$

— $n-3$ —

$$\leftarrow \underline{D(T_{n-1}, D(T_{n-2}, D(T_{n-3}, \mathit{and}(D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4})),} \quad \mathrm{MP}'_{n-3}$$
$$\underline{D(T_{n-4}, \psi_{n-4})))))}$$

$$\leftarrow \underline{D(T_{n-1}, (D(T_{n-2}, D(T_{n-3}, \mathit{and}(D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4})),} \quad \mathrm{MP}'_{n}$$
$$\underline{D(T_{n-4}, \psi_{n-4})))) \Leftarrow \xi))} \wedge D(T_{n-1}, \xi)$$

$$\leftarrow \underline{D(T_{n-1}, \mathit{and}(D(T_{n-2}, (D(T_{n-3}, \mathit{and}(D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4})),} \quad \mathrm{MP}'_{n-1}$$
$$\underline{D(T_{n-4}, \psi_{n-4}))) \Leftarrow \xi')), D(T_{n-2}, \xi')))}$$

$$\leftarrow \underline{D(T_{n-1}, D(T_{n-2}, (D(T_{n-3}, \mathit{and}(D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4})),} \quad \mathrm{ANDI}'_{n}$$
$$\underline{D(T_{n-4}, \psi_{n-4}))) \Leftarrow \xi')))}$$
$$\wedge D(T_{n-1}, D(T_{n-2}, \xi'))$$

$$\leftarrow \underline{D(T_{n-1}, D(T_{n-2}, \mathit{and}(D(T_{n-3}, D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4}))),} \quad \mathrm{ANDI}'_{n-2}$$
$$\underline{D(T_{n-3}, D(T_{n-4}, \psi_{n-4}))}))))$$

$$\leftarrow \underline{D(T_{n-1}, (D(T_{n-2}, \mathit{and}(D(T_{n-3}, D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4}))),} \quad \mathrm{MP}'_{n}$$
$$\underline{D(T_{n-3}, D(T_{n-4}, \psi_{n-4})))) \Leftarrow \xi''))} \wedge D(T_{n-1}, \xi'')$$

$$\leftarrow \underline{D(T_{n-1}, \mathit{and}(D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4})))),} \quad \mathrm{ANDI}'_{n-1}$$
$$\underline{D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, \psi_{n-4})))))}$$

$$\leftarrow D(T_{n-1}, D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, (\phi_{n-4} \Leftarrow \psi_{n-4}))))) \quad \mathrm{ANDI}'_{n}$$
$$\wedge D(T_{n-1}, D(T_{n-2}, D(T_{n-3}, D(T_{n-4}, \psi_{n-4}))))$$

— $n-4$ —

The next segment $n-5$ (between bars $n-4$ and $n-5$) of , $^{\mathcal{GOT}}$ consists of the following sequence of resolving 'clauses': MP'_{n-4}, MP'_{n}, MP'_{n-1}, ANDI'_{n}, MP'_{n-2}, MP'_{n}, ANDI'_{n-1}, ANDI'_{n}, ANDI'_{n-3}, MP'_{n}, MP'_{n-1}, ANDI'_{n}, ANDI'_{n-2}, MP'_{n}, ANDI'_{n-1}, and ANDI'_{n}. The subsequent segment $n-6$ consists of: MP'_{n-5}, MP'_{n}, MP'_{n-1}, ANDI'_{n}, MP'_{n-2}, MP'_{n}, ANDI'_{n-1}, ANDI'_{n}, MP'_{n-3}, MP'_{n}, MP'_{n-1}, ANDI'_{n}, ANDI'_{n-2}, MP'_{n}, ANDI'_{n-1}, ANDI'_{n}, ANDI'_{n-4}, MP'_{n}, MP'_{n-1}, ANDI'_{n}, MP'_{n-2}, MP'_{n}, ANDI'_{n-1}, ANDI'_{n}, ANDI'_{n-3}, MP'_{n}, MP'_{n-1}, ANDI'_{n}, ANDI'_{n-2}, MP'_{n}, ANDI'_{n-1}, and ANDI'_{n}.

Induction base. The general part of the proof below does not entail the following results which thus must be proved separately. The first result has partially been established by the sample , $\mathcal{GOT}$. It is that resolution steps exist for segments $n-1$ and $n-2$. The second result is that the following SLD-refutations exist. Segments $n-1$ and $n-2$ require the existence of SLD-refutations of MP'_n, MP'_{n-1}, and ANDI'_n, segment $n-3$ requires the existence of SLD-refutations of ANDI'_{n-1} and UE'_{n-1}.

The required SLD-refutations can be constructed as follows. Eliminating the universal quantifiers for clauses MP_n, ANDI_n, UE_n gives clauses

$$\begin{array}{ll} \mathrm{MP}'_n & D(T_{n-1},\phi_{n-1}) \leftarrow D(T_{n-1},(\phi_{n-1} \Leftarrow \psi_{n-1})) \wedge D(T_{n-1},\psi_{n-1}) \\ \mathrm{ANDI}'_n & D(T_{n-1},\mathit{and}(\phi_{n-1},\psi_{n-1})) \leftarrow D(T_{n-1},\phi_{n-1}) \wedge D(T_{n-1},\psi_{n-1}) \\ \mathrm{UE}'_n & D(T_{n-1},\phi_{n-1}) \leftarrow D(T_{n-1},(\chi_{n-1})[\psi_{n-1}]) \\ & \qquad\qquad \wedge\, \mathit{Subst}(\chi_{n-1},\psi_{n-1},\upsilon_{n-1},\phi_{n-1}) \end{array}$$

for arbitrary ϕ_{n-1}, ψ_{n-1}, χ_{n-1} and υ_{n-1}.

An SLD-refutation of MP'_{n-1}

$$\leftarrow D(T_{n-1},(D(T_{n-2},\phi_{n-2}) \Leftarrow \mathit{and}(D(T_{n-2},(\phi_{n-2} \Leftarrow \psi_{n-2})),D(T_{n-2},\psi_{n-2}))))$$

is obtained by two consecutive resolutions with UE'_n and a final resolution with MP_{n-1}

$$D(T_{n-1},(a)[(b)[(D(T_{n-2},a) \Leftarrow \mathit{and}(D(T_{n-2},(a \Leftarrow b)),D(T_{n-2},b)))]]) \leftarrow .$$

An SLD-refutation of ANDI'_{n-1}

$$\begin{aligned} \leftarrow D(T_{n-1},(D(T_{n-2},\mathit{and}(\phi_{n-2},\psi_{n-2})) \Leftarrow \mathit{and}(&D(T_{n-2},\phi_{n-2}),\\ &D(T_{n-2},\psi_{n-2})))) \end{aligned}$$

is obtained by two consecutive resolutions with UE'_n and a final resolution with ANDI_{n-1}

$$D(T_{n-1},(a)[(b)[(D(T_{n-2},\mathit{and}(a,b)) \Leftarrow \mathit{and}(D(T_{n-2},a),D(T_{n-2},b)))]]) \leftarrow .$$

Finally, an SLD-refutation of UE'_{n-1}

$$\begin{aligned} \leftarrow D(T_{n-1},(D(T_{n-2},\phi_{n-2}) \Leftarrow \mathit{and}(&D(T_{n-2},(\chi_{n-2})[\psi_{n-2}]),\\ &\mathit{Subst}(\chi_{n-2},\psi_{n-2},\upsilon_{n-2},\phi_{n-2})))) \end{aligned}$$

is obtained by four consecutive resolutions with UE'_n and a final resolution with UE_{n-1}

$$\begin{aligned} D(T_{n-1},(q)[(p)[(x)[(y)[(D(T_{n-2},q) \Leftarrow \mathit{and}(&D(T_{n-2},(x)[p]),\\ &\mathit{Subst}(x,p,y,q)))]]]]) \leftarrow . \end{aligned}$$

Induction hypothesis. $\Gamma^{\mathcal{GOT}}$ is established up to and including segment $n-k$, $k \geq 1$, which means that SLD-refutations of MP'_{n-l}, ANDI'_{n-l} and UE'_{n-l} for $1 \leq l \leq (k-1)$ exist in $\Gamma^{\mathcal{GOT}}$.

Induction step. Analysing the sample $\Gamma^{\mathcal{GOT}}$ shows the following. Consider the segment $n-(k+1)$ to be constructed. Segments $n-1$ to $n-(k+1)$ identify the resolution steps between an arbitrary goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, D(T_{n-(k+1)}, \phi_{n-(k+1)})) \ldots)$$

and the composite goal

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)}))) \ldots)\\ &\;\wedge D(T_{n-1}, \ldots D(T_{n-k}, D(T_{n-(k+1)}, \psi_{n-(k+1)})) \ldots).\end{aligned}$$

Segment $n-(k+1)$ begins with a resolution with MP'_{n-k}

$$\begin{aligned}D(T_{n-1}, \ldots D(T_{n-k}, (D(T_{n-(k+1)}, \phi_{n-(k+1)})&\\ \Leftarrow \mathit{and}(D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)})),&\\ D(T_{n-(k+1)}, \psi_{n-(k+1)})))) \ldots) \leftarrow&\end{aligned}$$

and realizing this requires a result already obtained with the segments up to and including $n-k$, i.e., the resolution steps between an arbitrary goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, \phi_{n-k}) \ldots)$$

and the composite goal

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, (\phi_{n-k} \Leftarrow \psi_{n-k})) \ldots)\\ &\;\wedge D(T_{n-1}, \ldots D(T_{n-k}, \psi_{n-k}) \ldots).\end{aligned}$$

Resolution with MP'_{n-k} gives the first resolvent of segment $n-(k+1)$, i.e.,

$$\begin{aligned}\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, \mathit{and}(D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)})),&\\ D(T_{n-(k+1)}, \psi_{n-(k+1)}))) \ldots)&\end{aligned}$$

and the segment is completed when the resolution steps are established between this goal and the composite goal

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)})))) \ldots)\\ &\;\wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, D(T_{n-(k+1)}, \psi_{n-(k+1)}))) \ldots).\end{aligned}$$

This can be accomplished by again exploiting the resolution steps identified by the segments up to and including $n-k$. At a certain choice point in these steps, the current resolvent is

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, (D(T_{n-k}, \phi_{n-k}) \Leftarrow \xi)) \ldots)\\ &\;\wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, \xi) \ldots)\end{aligned}$$

and resolution occurs with $\mathrm{MP}'_{n-(k-1)}$

$$D(T_{n-1}, \ldots D(T_{n-(k-1)}, (D(T_{n-k}, \phi_{n-k}) \Leftarrow \mathit{and}(D(T_{n-k}, (\phi_{n-k} \Leftarrow \psi_{n-k})), D(T_{n-k}, \psi_{n-k})))) \ldots) \leftarrow$$

to obtain the resolvent

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \mathit{and}(D(T_{n-k}, (\phi_{n-k} \Leftarrow \psi_{n-k})), D(T_{n-k}, \psi_{n-k}))) \ldots).$$

Now in the resolution steps identified by segments up to and including $n-k$, substitute ϕ_{n-k} in

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, \phi_{n-k}) \ldots)$$

for

$$\mathit{and}(D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)})), D(T_{n-(k+1)}, \psi_{n-(k+1)}))$$

and instead of resolving with $\mathrm{MP}'_{n-(k-1)}$ at the choice point in question, resolve with $\mathrm{ANDI}'_{n-(k-1)}$

$$D(T_{n-1}, \ldots D(T_{n-(k-1)}, (D(T_{n-k}, \mathit{and}(\alpha_{n-k}, \beta_{n-k})) \Leftarrow \mathit{and}(D(T_{n-k}, \alpha_{n-k}), D(T_{n-k}, \beta_{n-k})))) \ldots) \leftarrow$$

which is an alternative owing to the form of this particular ϕ_{n-k}. The resolvent will be

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \mathit{and}(D(T_{n-k}, D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)}))), D(T_{n-k}, D(T_{n-(k+1)}, \psi_{n-(k+1)})))) \ldots).$$

Compare the original and the new resolvent and note that *and* appears at the same position. Furthermore, note that for the original resolvent the remainder of the resolution steps identified by the segments up to and including $n-k$ only move the *and* outwards until obtaining the composite goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, (\phi_{n-k} \Leftarrow \psi_{n-k})) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-k}, \psi_{n-k}) \ldots).$$

Since the *and* appears at the same position in the new resolvent, obviously by the same resolution steps we will obtain the composite goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, D(T_{n-(k+1)}, (\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)})))) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-k}, D(T_{n-(k+1)}, \psi_{n-(k+1)})) \ldots).$$

Thus, the resolution steps required for segment $n-(k+1)$ are exactly those identified by the segments up to and including $n-k$ followed again by the same steps, with the only modifications in this second sequence that resolution with $\mathrm{MP}'_{n-(k-1)}$ is replaced by resolution with $\mathrm{ANDI}'_{n-(k-1)}$.

Before the proof of lemma 1 is completed, one thing remains. The proof exploits $\mathrm{MP}'_{n-(k-1)}$ and $\mathrm{ANDI}'_{n-(k-1)}$ which is justified since by the induction hypothesis these and $\mathrm{UE}'_{n-(k-1)}$ have SLD-refutations succeeding, respectively, in final resolutions with $\mathrm{MP}_{n-(k-1)}$, $\mathrm{ANDI}_{n-(k-1)}$, and $\mathrm{UE}_{n-(k-1)}$. Also, however, segment $n-(k+1)$ requires that MP'_{n-k} has an SLD-refutation succeeding in a final resolution with MP_{n-k}, (and segment $n-(k+2)$ will require that ANDI'_{n-k} and UE'_{n-k} have SLD-refutations succeeding, respectively, in final resolutions with ANDI_{n-k}, and UE_{n-k}). Consider the following schematic description of an SLD-refutation (in which the vertical ellipses stand for resolution steps between the goals)

$$\begin{array}{ll}
\leftarrow D(T_{n-1},\dots D(T_{n-k},\Lambda)\dots) & \\
\quad\vdots & \\
\leftarrow D(T_{n-1},\dots \mathit{and}(D(T_{n-k},(b)[\Lambda']), & \\
\qquad\qquad \mathit{Subst}(b,\Lambda',\upsilon^1_{n-k},\Lambda))\dots) & \mathrm{UE}'_{n-(k-1)} \\
\quad\vdots & \\
\leftarrow D(T_{n-1},\dots \mathit{and}(D(T_{n-k},(a)[(b)[\Lambda'']]), & \\
\qquad\qquad \mathit{Subst}(a,(b)[\Lambda''],\upsilon^2_{n-k},(b)[\Lambda']))\dots) & \mathrm{UE}'_{n-(k-1)} \\
\quad\vdots & \\
\leftarrow & \mathrm{MP}_{n-k} \\
 & (\text{or } \mathrm{ANDI}_{n-k}).
\end{array}$$

It gives the common pattern for SLD-refutations of MP'_{n-k} and ANDI'_{n-k}, and, if repeated twice, also of UE'_{n-k}, where, for each goal, Λ is

$$\begin{aligned}(D(T_{n-(k+1)},\phi_{n-(k+1)}) \Leftarrow \mathit{and}(&D(T_{n-(k+1)},(\phi_{n-(k+1)} \Leftarrow \psi_{n-(k+1)})),\\ &D(T_{n-(k+1)},\psi_{n-(k+1)})))\end{aligned}$$

and

$$\begin{aligned}(D(T_{n-(k+1)},\mathit{and}(\phi_{n-(k+1)},\psi_{n-(k+1)})) \Leftarrow \mathit{and}(&D(T_{n-(k+1)},\phi_{n-(k+1)}),\\ &D(T_{n-(k+1)},\psi_{n-(k+1)})))\end{aligned}$$

and

$$\begin{aligned}(D(T_{n-(k+1)},\phi_{n-(k+1)}) \Leftarrow \mathit{and}(&D(T_{n-(k+1)},(\chi_{n-(k+1)})[\psi_{n-(k+1)}]),\\ &\mathit{Subst}(\chi_{n-(k+1)},\psi_{n-(k+1)},\upsilon_{n-(k+1)},\phi_{n-(k+1)}))).\end{aligned}$$

Consider $\mathrm{UE}'_{n-(k-1)}$. In fact, $\mathrm{UE}'_{n-(k-1)}$ encompasses expressions

$$\begin{aligned}D(T_{n-1}, \ldots D(T_{n-(k-1)}, (D(T_{n-k}, \phi_{n-k}) \Leftarrow \\ \mathit{and}(D(T_{n-k}, (\chi_{n-k})[\psi_{n-k}]), \\ \mathit{Subst}(\chi_{n-k}, \psi_{n-k}, \upsilon_{n-k}, \phi_{n-k})))) \ldots)\end{aligned}$$

for arbitrary ϕ_{n-k}, ψ_{n-k}, χ_{n-k}, υ_{n-k}. For instance, ϕ_{n-k}, ψ_{n-k}, χ_{n-k}, and υ_{n-k} may, respectively, be Λ, Λ', b, and the term substituting b in Λ' to obtain Λ. Observe that $\mathrm{UE}'_{n-(k-1)}$ has the form

$$D(T_{n-1}, \ldots D(T_{n-(k-1)}, (\alpha \Leftarrow \beta)) \ldots).$$

The part of the SLD-refutation up to the first resolution with $\mathrm{UE}'_{n-(k-1)}$ appears thus (where $\alpha = D(T_{n-k}, \Lambda)$):

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \alpha) \ldots) \\ &\qquad \vdots \\ &\leftarrow \underline{D(T_{n-1}, \ldots D(T_{n-(k-1)}, (\alpha \Leftarrow \beta)) \ldots)} \wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, \beta) \ldots) \\ &\qquad \{\beta \mapsto \mathit{and}(D(T_{n-k}, (b)[\Lambda']), \mathit{Subst}(b, \Lambda', \upsilon^1_{n-k}, \Lambda))\} \\ &\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \mathit{and}(D(T_{n-k}, (b)[\Lambda']), \qquad \mathrm{UE}'_{n-(k-1)} \\ &\qquad\qquad \mathit{Subst}(b, \Lambda', \upsilon^1_{n-k}, \Lambda))) \ldots)\end{aligned}$$

Thus we need the result already obtained by the segments up to and including $n-(k-1)$, i.e., the resolution steps between an arbitrary goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \phi_{n-(k-1)}) \ldots)$$

and the composite goal

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, (\phi_{n-(k-1)} \Leftarrow \psi_{n-(k-1)})) \ldots) \\ &\ \wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, \psi_{n-(k-1)}) \ldots).\end{aligned}$$

After the first resolution with $\mathrm{UE}'_{n-(k-1)}$ the *and* must be moved outwards in

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \mathit{and}(D(T_{n-k}, (b)[\Lambda']), \mathit{Subst}(b, \Lambda', \upsilon^1_{n-k}, \Lambda))) \ldots)$$

to obtain the composite goal

$$\begin{aligned}&\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, (b)[\Lambda'])) \ldots) \\ &\ \wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, \mathit{Subst}(b, \Lambda', \upsilon^1_{n-k}, \Lambda))) \ldots).\end{aligned}$$

Thus, as earlier in the proof we need to exploit the result already obtained in segments up to and including $n-k$, i.e., the resolution steps between an arbitrary goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, and(\phi_{n-(k-1)}, \psi_{n-(k-1)})) \ldots)$$

and the composite goal

$$\begin{aligned} &\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, \phi_{n-(k-1)}) \ldots) \\ &\wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, \psi_{n-(k-1)}) \ldots). \end{aligned}$$

Now consider the composite goal

$$\begin{aligned} &\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, (b)[\Lambda'])) \ldots) \\ &\wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, D(T_{n-k}, Subst(b, \Lambda', v^1_{n-k}, \Lambda))) \ldots). \end{aligned}$$

The second subgoal is assumed to succeed by resolution with a unit clause encoding permitted substitutions for theory T_{n-k}. Repeating the above steps yields the resolution steps between the first goal and the next resolution with $\mathrm{UE}'_{n-(k-1)}$. Now, after moving *and* outwards, resolution will occur with the unit clause MP_{n-k} and the SLD-refutation of MP'_{n-k} succeeds. (Equally, preparatory to segment $n-(k+2)$, the same resolution steps as those composing the SLD-refutation of MP'_{n-k} will give the SLD-refutation of ANDI'_{n-k} succeeding in a final resolution with ANDI_{n-k}, and repeating them twice gives the SLD-refutation of UE'_{n-k} succeeding in a final resolution with UE_{n-k}.)

This completes the proof of lemma 1. ∎

Lemma 2. *Every SLD-resolution step of* $\Gamma^{\mathcal{LOT}}$

$$\begin{aligned} &\leftarrow D(T_{n-i}, and(\phi_{n-i}, \psi_{n-i})) \\ &\leftarrow D(T_{n-i}, \phi_{n-i}) \wedge D(T_{n-i}, \psi_{n-i}) \quad \text{(by Case 2)} \end{aligned}$$

can be constructed in $\Gamma^{\mathcal{GOT}}$ *by corresponding steps*

$$\begin{aligned} &\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, and(\phi_{n-i}, \psi_{n-i})) \ldots) \\ &\quad \vdots \\ &\leftarrow D(T_{n-1}, \ldots D(T_{n-i}, \phi_{n-i}) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-i}, \psi_{n-i}) \ldots). \end{aligned}$$

Proof. By lemma 1, resolution steps exist between a goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, and(\phi_{n-k}, \psi_{n-k})) \ldots)$$

and the composite goal

$$\begin{aligned} &\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, (and(\phi_{n-k}, \psi_{n-k}) \Leftarrow \xi)) \ldots) \\ &\wedge D(T_{n-1}, \ldots D(T_{n-k}, \xi) \ldots). \end{aligned}$$

An SLD-refutation of the first subgoal exists by resolution with $\mathrm{ANDI}'_{n-(k-1)}$, producing the substitution

$$\xi \mapsto and(D(T_{n-k}, \phi_{n-k}), D(T_{n-k}, \psi_{n-k}))$$

and the second subgoal becomes

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, and(D(T_{n-k}, \phi_{n-k}), D(T_{n-k}, \psi_{n-k}))) \ldots).$$

Again by lemma 1, resolution steps exist between a goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, and(D(T_{n-k}, \phi_{n-k}), D(T_{n-k}, \psi_{n-k}))) \ldots)$$

and the composite goal

$$\begin{aligned} &\leftarrow D(T_{n-1}, \ldots D(T_{n-(k-1)}, (and(D(T_{n-k}, \phi_{n-k}), D(T_{n-k}, \psi_{n-k})) \Leftarrow \xi')) \ldots) \\ &\wedge D(T_{n-1}, \ldots D(T_{n-(k-1)}, \xi') \ldots) \end{aligned}$$

and an SLD-refutation of the first subgoal exists by resolution with $\mathrm{ANDI}'_{n-(k-2)}$, etc.

By repeating this k times the *and* is moved outwards so that resolution with ANDI'_n may occur, and the final resolution steps will be between

$$\begin{aligned} \leftarrow D(T_{n-1}, and(&D(T_{n-2}, \ldots D(T_{n-k}, \phi_{n-k}) \ldots), \\ &D(T_{n-2}, \ldots D(T_{n-k}, \psi_{n-k}) \ldots))) \end{aligned}$$

and the composite goal

$$\leftarrow D(T_{n-1}, \ldots D(T_{n-k}, \phi_{n-k}) \ldots) \wedge D(T_{n-1}, \ldots D(T_{n-k}, \psi_{n-k}) \ldots). \quad \blacksquare$$

3.4.3. Limitations of the Local Characterization. Above, the structure of $\mathcal{IT}$ has been described from a particular level n. Specifying the theory of primary rules T_1 metatheoretically in terms of what can be proved from T_1 may seem natural. The topmost theory T_n gives an axiomatic definition of provability of theory T_{n-1} and indirectly of all theories T_i, $i < n$, hence, embracing all the non-logical axioms of these theories. Based on the idea of local characterization, the multilayered theory T_n can be directly encoded in a Horn clause program as illustrated in our first study [34]. In that study, however, we intentionally ignored the problem of furnishing external knowledge, a problem which calls for another implementation approach. That is, both the global and local characterization above simply presuppose the availability of all non-logical axioms needed for completing a computation.

Thus, the approach needs amplification to cope with assimilation of non-logical axioms. Described briefly, our amplified approach is as follows. The hierarchical structure of the formal object theory $\mathcal{OT}$ representing $\mathcal{IT}$ is taken as a composite object language which can be characterized in a theory $\mathcal{MT}$ of a separate metalanguage. This metalanguage thus takes the whole n-level language of $\mathcal{OT}$ as its object language. For example, the formula

$$Prover(Demo(n(T_1), n(A)), \ldots, \ldots, Proof)$$

expresses in $\mathcal{MT}$, that in the formal theory of $\mathcal{OT}$, which represents the informal theory T_1 of $\mathcal{IT}$, a formalization of a primary rule A is included. In the metalanguage this inclusion is verified by a sequence *Proof* of statements each of which names an 'object/object'-inference or a 'meta/object'-inference of the object language, thus constituting a metaproof in $\mathcal{MT}$ of a formal proof in $\mathcal{OT}$. $\mathcal{MT}$ is a semiformal interactive metalogic program encoding knowledge for proposing meaningful formulas of $\mathcal{OT}$ and simulating the inferencing in $\mathcal{OT}$ for accepting or rejecting these formulas. The next chapter describes $\mathcal{MT}$ in detail.

A system corresponding directly to $\mathcal{LOT}$ would be impractical. Answers to the user's queries would be restricted to 'yes' and 'no'. In order to access supplementary information, such as which rules (instances of schemata) were involved in the reasoning, it would be necessary to trace the computation of queries. For assessing the adequacy of these rules, the user would have to extract important information from the trace, e.g., from which rule schemata the rules have been specialized and in what legal context, whether the proposed specializations of the gaps in these schemata belong to accurate types, etc. Moreover, additional information such as the textual representation of rule schemata would be unavailable. Obviously, these are aspects that the computer should support and $\mathcal{MT}$ incorporates facilities of this kind.

CHAPTER 4

THE SEMIFORMAL METATHEORY

This chapter analyses formalization of inexact in contrast to exact theories. A semiformal theorem prover is presented that interactively proposes formal theories corresponding to possible interpretations of the informal theory.

4.1. METALOGIC AND SEMIFORMALIZATION

We now describe our metatheory $\mathcal{MT}$, mainly as a program partially characterizing $\mathcal{OT}$ whose intended interpretation is $\mathcal{IT}$. Our metalogic consists of Horn clauses and the inference mechanism of Prolog. In this metalanguage $\mathcal{MT}$ is represented by formulas, i.e., in Horn clauses, and the language of $\mathcal{OT}$ by terms, i.e., terms of Horn clauses. All variables of our metalanguage, i.e., the language of $\mathcal{MT}$, are metavariables denoting names of linguistic objects of the object language, i.e., the language of $\mathcal{OT}$; often we use in the metalanguage the symbols of the object language autonymously.

To the *formation rules* and *rules of inference* (including axiom schemata) of proof theory, the *rules of meaningfulness* and the *rules of acceptance*, respectively, correspond. Both have a more vague character than their proof theory counterparts.

It is a difficult and sometimes impossible problem to state formation rules in the metatheory which can cope with the vagueness of legal concepts. Therefore, in contrast to proof theory it seems necessary to complement with an external assessment of which rules are meaningful rules, and which rules are acceptable as true material implications.

In general, the specification of a formal theory is done from the starting-point of an informal theory which is the intended interpretation the formalism should capture. For example, a constant in the formal language of the formal theory is assumed to have a natural counterpart in the informal theory, e.g., to denote a specific individual or a specific category of individuals.

In our case, however, we cannot predetermine this connection between a formal and an informal theory, because we cannot decide in advance all legal concepts that can be relevant in a legal system. This is something that must be resolved from case to case. Therefore, we leave it to the user to decide the formal counterpart of his informal understanding of a legal concept, e.g., an informal legal concept 'hirer' of $\mathcal{IT}$ will naturally get the constant 'hirer' of the formal language of $\mathcal{OT}$. That is to say, we presuppose an immediate or autonymous relation between a symbol or symbols of the formal language and its informal counterpart. Thus, with external help a metatheory $\mathcal{MT}$ could be extended and some of the gaps in its schemata for non-logical axioms of $\mathcal{OT}$ be filled to yield a formalization of rules of $\mathcal{IT}$ obtained by specializing the available vague descriptions of these rules. In that way, a program representing $\mathcal{MT}$ could construct, interactively with the user, a metaproof in $\mathcal{MT}$ representing a formal proof in $\mathcal{OT}$; at least for the proposal of a legal case under consideration. How adequately these formal proofs, represented in the metatheory $\mathcal{MT}$, correspond to $\mathcal{IT}$ is a matter for external assessment.

In some sense this position could be understood as performing a specification in a metatheory of the formal language of a formal theory as well as deciding its non-logical axioms at the same time as drawing some conclusion there from. At the end of a session (of constructing metaproofs) the relevant sections of this theory, and the formal proof of a particular legal case could be presented to the user for a final examination by extracting the formal proof from the metaproof. Observe that $\mathcal{MT}$ is a *semiformal* theory in the sense that it has both a formal part, consisting of sentences represented as Horn clauses, and an informal part consisting of user interpreted sentences.

Thus, the *rules of meaningfulness* in $\mathcal{MT}$ can only partially characterize sentences of the object language, i.e., the language of $\mathcal{OT}$. In the representation in the metalanguage, we have to assume a fixed structure for designating a class of rules, i.e., a schema. Within this structure, local differences must be met, i.e., different specializations of the schema have to give different representations of sentences (rules) of the object language. These local differences are expressed by metavariables which have to be filled in by a user and satisfy certain interactively investigated typing conditions. This is illustrated in the program clause below.

For the clauses below we postulate an inverse law of naming, i.e., `n(A)=n(B)` $\rightarrow$ `A=B`. As to the problem of naming in metaprogramming, observe that all variables are metavariables; there are no object variables. As described in Sect. 3.4 this is consonant with the view that $\mathcal{IT}$ contains only ground rules besides schemata and rules of logic. The variables range over linguistic expressions of the languages and that they become bound to objects of the correct sort is furthered by our typing guidelines. Together with the user, $\mathcal{MT}$ suggests that $\mathcal{IT}$ contains a certain instantiated rule applying to the current case by defining $\mathcal{OT}$ as containing a certain ground formula representing the rule.

```
meaningful_sent(t(1),RuleProp1,[ModAt1,unspec],LegSet1,Text):-
  RuleProp1 = (legal_cons(pay,X,Y,goods,price)<=
                    and(actor_1(X,goods),
                    and(actor_2(Y,goods),
                    and(unsettled_price(goods),
                    and(demands(Y,price),
                    reasonable(price,goods)))))),
```

```
  ModAt1 = [X/vendee,Y/vendor], Types = [actor/X,actor/Y],
  LegSet1 = [[provisionno(sga(5))|_],LegSet0],
  Text = 'If a sale of goods has been made but no price settled
          then the vendee should pay what the vendor demands if
          reasonable. ',
  accurate_typing(t(1),RuleProp1,ModAt1,Types,Text).
```

The program clause characterizes the provision Sect. 5 Sale of Goods Act, i.e., the rule schema 1 in Fig. 2.1, which describes a class of rules of $\mathcal{IT}$. Its linguistic wording is specified in the metavariable `Text`. In $\mathcal{IT}$ this provision is assumed to be open with respect to the concepts 'vendee' and 'vendor'. So, its assumed fixed structure is represented in the metalanguage as the term specified for the metavariable `RuleProp1` with gaps expressed by the metavariables `X` and `Y`. These variables have to be specialized interactively with the user and the predicate `accurate_typing/5` is defined for this interaction. The metavariable `ModAt1` expresses the relation between the concepts of $\mathcal{IT}$, i.e., the text of `Text` and its open parts, i.e., 'vendee' and 'vendor', and its formal counterpart in $\mathcal{OT}$ partly specified in `RuleProp1`. Thus, accurate typing carried out by the user gives a meaningful rule of the object language of level 1, represented in the metalanguage by the specialized term of `RuleProp1`. The metavariable `LegSet1` identifies which part of level 1 in $\mathcal{IT}$ is relevant for a particular case.

The *rules of acceptance* also may be only partially characterized in the metalanguage. However, a user can interactively add interpretation data, thereby extending the partial characterization of $\mathcal{OT}$ in the theory $\mathcal{MT}$ of the metalanguage. What is hard to characterize is the determination of whether or not a meaningful rule belongs to a theory of $\mathcal{IT}$, i.e., is legally acceptable, and thus should have a formal counterpart in $\mathcal{OT}$. At present, this is solved by assuming in $\mathcal{MT}$ that a rule is acceptable when a user tries to apply it, and the conditions for its application are accepted, i.e., either follow by logic from other accepted rules or are included in the theory by rules at the higher adjacent level in co-operation with the user. So, we presuppose that it is only the user who can determine the relevance of a specific principle. Consequently, at the end of a session it should be possible for a user to examine these assumptions. Let us illustrate this with a program clause.

```
prover(demo(n(t(I)),n(SentProp_I)),Mod_I,LegSet_I,Proof_I):-
  propose_sent(t(I),SentProp_I,Mod_I,LegSet_I),
  J is I + 1,
  ground([SentProp_I,Mod_I,LegSet_I]),
  permissible(t(I),SentProp_I),
  prover(demo(n(t(J)),n(demo(n(t(I)),n(SentProp_I)))),
         [ModAt_J,Mod_I],[LegSetAt_J,LegSet_I],Proof_J),
  Proof_I = (sentence_of(theory(I),SentProp_I)<=
               proof_of(theory(J),
                        proved(theory(I),SentProp_I),
```

```
Proof_J)).
```

This metatheory Horn clause states about the formalization $\mathcal{OT}$ of theories T_i and T_j at adjacent levels i and j in the informal theory $\mathcal{IT}$: for rules named `n(SentProp_I)` modified by `Mod_I` in the legal setting `LegSet_I` there is an $\mathcal{MT}$ proof `Proof_I` that $T_i \vdash \mathit{SentProp}_i$, if there is an $\mathcal{MT}$ proof `Proof_J` to the effect that $T_j \vdash$ '$T_i \vdash \mathit{SentProp}_i$' with appropriate modifiers and further specification of the legal context. The predicate `propose_sent`/4 is defined to specialize interactively with a user a `meaningful_sent` schema in a legal context. Both `Mod_I` and `LegSet_I` should be stored in `Proof_I` together with the text of `SentProp_I` for later use when the proof is displayed and verified interactively. Here we exclude this information for typographical reasons.

That the layering of $\mathcal{IT}$ is potentially infinite—or at least embraces arbitrary many levels—can be preserved in a program defining its formalization $\mathcal{OT}$. The program's terminating condition must depend on some topmost level but this need not be predetermined but can instead be identified, e.g., as the first level lacking applicable rules.

This gradual definition in metalogic of the formal object theory $\mathcal{OT}$, representing the informal legal theory $\mathcal{IT}$, can be seen as a knowledge-based system with a user interaction level where, for instance, a particular factual situation can be transformed into a legal case, i.e., a proposed primary rule. That is to say, the factual situation is interpreted legally and formalized. Its particular objects and relations expressed in colloquial natural language are translated into legal terminology and it is given a formal counterpart on level 1. This interpretation process is assumed to result in a primary rule proposal that has to be validated by the upper levels in the legal reasoning process. Moreover, one important point worth emphasizing concerning a computer system supporting legal reasoning is that the derived legal reasoning can be displayed and thereby ultimately validated by the user. Another important point is that the system should allow the user to experiment in a flexible way with multiple interpretations of a factual situation in order to reveal the consequences of several plausible understandings often only differing in minor details. Our program implementing the metatheory will eventually display the legal reasoning arrived at, so that the user can determine whether or not the assumed rules are really true in the user's model of the problem (the informal theory $\mathcal{IT}$).

Hitherto the theoretical analysis has occasionally been supplemented with fragments of program code from the actual implementation of $\mathcal{MT}$. In the sequel the implementation program and its execution is more thoroughly accounted for.

4.2. A SEMIFORMAL METALOGIC THEOREM PROVER

The core of the implementation program is the `prover` clauses which constitute a semiformal metalogic theorem prover. They belong to $\mathcal{MT}$ which takes as object theory the entire multilayered $\mathcal{OT}$. The head of a `prover` clause appears schematically as

```
prover(demo(n(t(I)),n(A)),...,...,Proof)
```

where the first `demo` argument defines the formalization in $\mathcal{OT}$ of logic provability between a theory T_i of $\mathcal{IT}$ and a sentence of $\mathcal{IT}$ but though e.g., the fourth proof term argument has a counterpart in $\mathcal{OT}$—a formal proof extending over the whole hierarchy of $\mathcal{OT}$—it includes expressions solely of $\mathcal{MT}$ as well. As regards the reading of the first argument of `demo`, recall that T_i (with formal counterpart `t(I)` in $\mathcal{OT}$) is taken as a collection of the formulas at level i in $\mathcal{IT}$, which, at level $i+1$ in $\mathcal{IT}$, is named by $n(T_i)$ (with formal counterpart `n(t(I))` in $\mathcal{OT}$), cf. p. 15.

Clause `[ANDI]` deals with $\wedge$-introduction. In $\mathcal{MT}$ a theory T_i of $\mathcal{IT}$, with `t(I)` as formal counterpart in $\mathcal{OT}$, is assumed to include a sentence which is a conjunction if both its conjuncts may be assumed to be included in T_i. `ProofI` is a metaproof in $\mathcal{MT}$ of the existence of a sequence of formulas in $\mathcal{OT}$'s formalization of $\mathcal{IT}$ constituting a formal proof of the sentence, which, as explained below, is only a proposed sentence.

```
[ANDI]
prover(demo(n(t(I)),n(and(G1,G2))),
       [[ModG1,ModG2],ModsBelow],LegSetI,ProofI):-
  I ≥ 2,
  prover(demo(n(t(I)),n(G1)),[ModG1,ModsBelow],LegSetI,ProofG1),
  prover(demo(n(t(I)),n(G2)),[ModG2,ModsBelow],LegSetI,ProofG2),
  ProofI =(sentence_of(theory(I),and(G1,G2))<=
             and(proof_of(theory(I),G1,ProofG1),
             proof_of(theory(I),G2,ProofG2))).
```

Clause `[MP]` encodes *modus ponens*. In $\mathcal{MT}$ a theory T_i of $\mathcal{IT}$, with formal counterpart `t(I)` in $\mathcal{OT}$, is assumed to include a sentence which is the consequence of a proposed implication of T_i, whose content has been accepted, and whose antecedent can be assumed to be included in T_i and thus also shown to have acceptable content. In $\mathcal{MT}$, `LegSetI` and `ModI` identify and modify formula schemata corresponding to known fragments of sentences of the theory T_i. The predicate `propose_sent`/4 is defined to specialize interactively with a user such `meaningful_sent` schemata.

```
[MP]
prover(demo(n(t(I)),n(HeadI)),ModI,LegSetI,ProofI):-
  I ≥ 2,
  propose_sent(t(I),(HeadI<=BodyI),ModI,LegSetI),
  prover(demo(n(t(I)),n(BodyI)),ModI,LegSetI,ProofBodyI),
  ProofI = (sentence_of(theory(I),HeadI)<=
              and(rule_of(theory(I),(HeadI<=BodyI)),
              proof_of(theory(I),BodyI,ProofBodyI))).
```

Clause [UP] (short for upward reflection) encodes in $\mathcal{MT}$ the formalization in $\mathcal{OT}$ of upward reflection between two theories T_i and T_j of arbitrary adjacent levels in $\mathcal{IT}$, with formal counterparts t(I) and t(J) in $\mathcal{OT}$. A sentence is assumed to belong to a theory T_i if this accords with the rules of theory T_j of the higher adjacent level.

```
[UP]
prover(demo(n(t(I)),n(SentPropI)),ModI,LegSetI,ProofI):-
  propose_sent(t(I),SentPropI,ModI,LegSetI),
  J is I + 1,
  ground([SentPropI,ModI,LegSetI]),
  permissible(t(I),SentPropI),
  prover(demo(n(t(J)),n(demo(n(t(I)),n(SentPropI)))),
         [ModAtJ,ModI],[LegSetAtJ,LegSetI],ProofJ),
  ProofI = (sentence_of(theory(I),SentPropI)<=
              proof_of(theory(J),
                       proved(theory(I),SentPropI),
                       ProofJ)).

permissible(t(I),SentPropI):-I = 1.
permissible(t(I),SentPropI):-I ≥ 2,\+ SentPropI = (Head<=Body).
```

As explained on p. 16, upward reflection must be constrained. If each sentence were upward reflected directly when proposed, the reasoning process would ascend directly to the topmost level since the metarule proposed for assessing the sentence would itself directly be upward reflected, etc. Therefore, at levels i, $i \geq 2$, only sentence proposals which are ground facts may be upward reflected, thus postponing the assessment of rules, which may only be proposed as non-ground conditional sentences, till facts are activated by their premises. According to this reasoning scheme the content of all sentences involved in the reasoning process will eventually be assessed. The restriction is maintained by the permissible subgoal.

For assessing sentence proposals for the adjacent lower level theory T_i the knowledge of rules in $\mathcal{IT}$ will at some level j be too rudimentary for composing a theory T_j. At this level, T_j is considered to be the user's opinion of the sentences proposed for T_i. This is encoded in $\mathcal{MT}$ in the clause [TOP].

```
[TOP]
prover(demo(n(t(J)),n(demo(n(t(I)),n(RulePropI)))),
       ModJ,LegSetJ,ProofJ):-
  J ≥ 2,
% \+ propose_sent(t(J),(demo(n(t(I)),n(RulePropI))<=BodyJ),
%                   ModJ,LegSetJ),[1]
  external_confirmation(t(I),RulePropI,ModJ,LegSetJ),
  ProofJ = externally_confirmed(sentence_of(theory(I),RulePropI)).
```

Note that the topmost level is not fixed but dynamically identified as the first level for which no schemata exist from which suitable rules can be proposed to pursue the computation of the current goal.

4.3. A COMPUTATION OF A SAMPLE QUERY

Below, a partial trace of the computation of a sample query is presented, supplemented with program code necessary for its understanding. Here we present the internal execution but later in Ch. 6 the corresponding user interaction session is described and analysed. The query is

```
prover(demo(n(t(1)),n(RuleProp1)),Mod1,LegSet1,Proof1).
```

This query could be read as 'is there a metaproof `Proof1` stating that the theory T_1 of level 1 in $\mathcal{IT}$ includes a primary rule which is formalized in $\mathcal{OT}$ by `RuleProp1` and modified by `Mod1` in the legal setting `LegSet1`?' Since it is completely unspecified at this point which particular problem to solve, the query can be stated in these general terms and be generated by the system.

The goal resolves with the prover clause `[UP]` leading to six subgoals, the last of which constructs the proof term to bind `Proof1`. Below, we refrain from discussing how the proof term is constructed during the computation. The first subgoal of `[UP]` is

```
propose_sent(t(1),SentProp1,Mod1,LegSet1)
```

which (as will be described in the session presented below in Ch. 6) through user interaction selects a legal rule and modifies it for the current case.

The unifying clause

```
propose_sent(Theory,RuleProp,Mod,LegSet):-
  (Theory = t(1);RuleProp = (demo(_,_)<=Body)),[2]
  find_legal_setting(Theory,LegSet),
  meaningful_sent(Theory,RuleProp,Mod,LegSet,Text).
```

identifies the relevant part of the legal domain from which it retrieves a proposal for a rule, provided it is meaningful. The latter is sorted out by `meaningful_sent` clauses, such as e.g.,

```
meaningful_sent(t(1),RuleProp1,[ModAt1,unspec],LegSet1,Text):-
  RuleProp1 = (legal_cons(pay,X,Y,goods,price)<=
                  and(actor_1(X,goods),
```

[1] Removing the `%`'s before this subgoal gives the clause a declarative reading, but also superfluous questions. Therefore, the implemented program relies instead on the order of the `prover` clauses.

[2] The legal setting may be assumed to be unknown if either of these two conditions applies.

```
                and(actor_2(Y,goods),
                and(unsettled_price(goods),
                and(demands(Y,price),
                reasonable(price,goods)))))),
  ModAt1 = [X/vendee,Y/vendor], Types = [actor/X,actor/Y],
  LegSet1 = [[provision_no(sga(5))|_],LegSet0],
  Text = 'If a sale of goods has been made but no price settled
         then the vendee should pay what the vendor demands if
         reasonable. ',
  accurate_typing(t(1),RuleProp1,ModAt1,Types,Text).
```

The `accurate_typing` condition is intended to promote that user proposed modifications preserve the rule's meaningfulness.

Suppose now that the result of the user interaction is that the first subgoal of `[UP]` returns with the following ground argument bindings, i.e., the schema from Sect. 5 Sale of Goods Act is adapted into a primary rule proposal regulating a case of 'hire of goods' (where ⟨*name*⟩ is shorthand for an occurrence of the term named by *name*).

```
LegSet1 =
   [[provision_no(sga(5)),
     provision_category('Determination of Purchase-Money'),
     legal_field('Commercial Law')],unspec]          call it ⟨leg_set_1⟩
Mod1 = [[hirer/vendee,letter/vendor],unspec]           call it ⟨mod_1⟩
RuleProp1 =
   (legal_cons(pay,hirer,letter,goods,price)<=
      and(actor_1(hirer,goods),
      and(actor_2(letter,goods),
      and(unsettled_price(goods),
      and(demands(letter,price),
      reasonable(price,goods))))))                 call it ⟨rule_prop_1⟩
```

Now it must be established whether it accords with the higher adjacent level, i.e., the theory T_2, to assume a primary rule with this proposed content is included in the theory T_1. This is accomplished through 'upward reflection'.

Before a formula with content information is upward reflected it must be checked for groundness and permissibility (we must admit our procedure for `ground` is only a hack). These are the tasks of the third subgoal of `[UP]`,

`ground([`⟨*rule_prop_1*⟩`,`⟨*mod_1*⟩`,`⟨*leg_set_1*⟩`])` ,

and of the fourth subgoal of `[UP]`,

`permissible(t(1),`⟨*rule_prop_1*⟩`)` ,

which permit a conditional rule on level 1 to be upward reflected. The fifth, 'upward reflection', subgoal of `[UP]`

```
prover(demo(n(t(2)),n(demo(n(t(1)),n(⟨rule_prop_1⟩))))),
       [ModAt2,⟨mod_1⟩],[LegSetAt2,⟨leg_set_1⟩],Proof2),
```

resolves with the prover clause `[MP]` leading to four subgoals (the first and last of which, respectively, controls the index of the current level and constructs the proof term).

Now a secondary rule must be proposed with which to assess the lower level expression. The second subgoal of `[MP]` is

```
propose_sent(t(2),(demo(n(t(1)),n(⟨rule_prop_1⟩))<=Body2),
             ⟨mod_2⟩,⟨leg_set_2⟩),
```

where

⟨*mod_2*⟩ is `[ModAt2,⟨mod_1⟩]`,
⟨*mod_1*⟩ is `[⟨mod_at_1⟩,unspec]`,
⟨*mod_at_1*⟩ is `[hirer/vendee,letter/vendor]`, and
⟨*leg_set_2*⟩ is `[LegSetAt2,⟨leg_set_1⟩]`.

Suppose the user chooses the *analogia legis* principle. The relation between primary rules of theory T_1 and secondary rules for *analogia legis* of theory T_2 is encoded in this clause:

```
meaningful_sent(t(2),RuleProp2,Mod2,LegSet2,Text):-
  RuleProp2 =
    (demo(n(t(1)),n(RuleProp1))<=
       analogia_legis(n(RuleProp1),n(ModAt1),LegSet1)),
  Mod2 = [_,[ModAt1,_]],
  LegSet2 = [[interpretation_theory('analogia legis')|_],LegSet1],
  Text = '"A primary rule proposal is legally valid (i.e.,
           belongs to the theory t1 of valid primary rules)
           if its inclusion accords with the secondary rule for
           analogia legis."...',
  accurate_typing(t(2),RuleProp2,[],[],Text).
```

The second subgoal of `[MP]` returns with its second argument bound to

```
(demo(n(t(1)),n(⟨rule_prop_1⟩))<=
   analogia_legis(n(⟨rule_prop_1⟩),n(⟨mod_at_1⟩),⟨leg_set_1⟩))
```

and `LegSetAt2` bound to `[interpretation_theory('analogia legis')]`.

The third subgoal of `[MP]` is

```
prover(demo(n(t(2)),
       n(analogia_legis(n(⟨rule_prop_1⟩),
                        n(⟨mod_at_1⟩),⟨leg_set_1⟩)))),
       ⟨mod_2⟩,⟨leg_set_2⟩,ProofBody2),
```

which recursively calls [MP]. Now a meaningful proposal for an actual *analogia legis* secondary rule will, by the second propose_sent subgoal of [MP], be retrieved from this clause

```
meaningful_sent(t(2),RuleProp2,_,LegSet2,Text):-
  RuleProp2 =
    (analogia_legis(n((Cons<=Ante)),n(ModAt1),LegSet1)<=
       and(not(casuistical_interpretation(LegalField,
                                          n((not(Cons)<=Ante)))),
       and(intended_for(ProvisionNo,n(TypeCase)),
       and(substantial_similarity(n(TypeCase),n(Ante),n(ModAt1)),
       and(intended_to_meet(ProvisionNo,Interests,LegalField),
       and(supports(ProvisionNo,n(ModAt1),ProInt,Interests),
       and(recommend_rejection(ProvisionNo,n(ModAt1),
                               ContraInt,Interests),
       outweigh(ProInt,ContraInt)))))))),
  LegSet2 = [[interpretation_theory('analogia legis')|_],LegSet1],
  LegSet1 = [[provision_no(ProvisionNo),_,
             legal_field(LegalField)],_],
  Text = 'A certain rule may be applied to a case not subsumed,
          or at least not with certainty subsumed, under the
          rule's linguistic wording, if the case is not the
          object of a particular explicit rule, if the case has
          a substantial similarity to those the rule is intended
          for, if interests of some importance, which the rule
          is intended to meet, support such an application, and
          if no contrary interests exist recommending the
          rejection of such an application.',
  accurate_typing(t(2),RuleProp2,[],[],Text).
```

with these bindings (where ⟨*rule_prop_1*⟩ is (⟨*cons_rule_1*⟩<=⟨*ante_rule_1*⟩))

```
analogia_legis(n((⟨cons_rule_1⟩<=⟨ante_rule_1⟩)),
               n(⟨mod_at_1⟩),⟨leg_set_1⟩)<=
  and(not(casuistical_interpretation('Commercial Law',
                                     n((not⟨cons_rule_1⟩<=
                                            ⟨ante_rule_1⟩))))),
  and(intended_for(sga(5),n(TypeCase)),
  and(substantial_similarity(n(TypeCase),n(⟨ante_rule_1⟩),
                             n(⟨mod_at_1⟩))),
  and(intended_to_meet(sga(5),Interests,'Commercial Law'),
  and(supports(sga(5),n(⟨mod_at_1⟩),ProInt,Interests),
  and(recommend_rejection(sga(5),n(⟨mod_at_1⟩),
                          ContraInt,Interests),
  outweigh(ProInt,ContraInt))))))).
```

Now it must be proved that with the proposed content the antecedent of the *analogia legis* rule (call it ⟨*al_body*⟩) is included in T_2. The third subgoal of [MP] is

```
prover(demo(n(t(2)),n(⟨al_body⟩)),_,
       [[interpretation_theory('analogia legis')],⟨leg_set_1⟩],_),
```

and each of the conjuncts in ⟨*al_body*⟩ will be demonstrated in turn by the prover clauses [ANDI], [MP], and [UP]. To illustrate how user proposed content for a sentence is accepted (or rejected) at higher levels, let us focus on the fourth conjunct which gives rise to the goal

```
prover(demo(n(t(2)),
            n(intended_to_meet(sga(5),
                               Interests,
                               'Commercial Law'))),
       _,
       [[interpretation_theory('analogia legis')|_],⟨leg_set_1⟩]).
```

An 'intended to meet' sentence must be proposed by the user. The result may be a meaningful fact (unconditional sentence) whose inclusion in the theory T_2 must be accepted by the rules of theory T_3 or it may be a rule (conditional sentence) which is assumed to be included in T_2 directly after the user's acceptance. The resolving clauses in the respective cases are [UP] and [MP]. Thus, in the first case upward reflection occurs immediately. In the second case upward reflection is postponed until backward inferencing by *modus ponens* at the current level leads to the proposal of a fact. Note that this guarantees that the application of the originally proposed rule is not accepted unless all the components of its antecedent are eventually assessed and accepted.

Suppose a fact is proposed. The goal will resolve with the prover clause [UP] giving rise to six subgoals (the last of which constructs the proof term).

The first subgoal of [UP] is

```
propose_sent(t(2),
             intended_to_meet(sga(5),Interests,'Commercial Law'),
             [ModAt_2,⟨mod_1⟩],
             [[interpretation_theory('analogia legis')],
              ⟨leg_set_1⟩]).
```

Suppose it succeeds with bindings

```
ModAt_2 = ['hirer protection'/'consumer protection')]
Interests = 'hirer protection',
```

i.e., in the text below of the 'intended to meet' rule

```
'The provision Sect. 5 Sale of Goods Act is to be interpreted as
if it were intended to protect consumers and similar groups. ',
```

the concept `Interests` is exemplified by `'consumer protection'`, which has now been substituted for `'hirer protection'`.

Subgoals two and three of `[UP]` increase the level index by one and check whether

```
[intended_to_meet(sga(5),'hirer protection','Commercial Law'),
 ['hirer protection'/'consumer protection'],
 [[interpretation_theory('analogia legis')],⟨leg_set_1⟩]]
```

is ground preparatory to the fifth upward reflection subgoal of `[UP]`

```
prover(demo(n(t(3)),
            n(demo(n(t(2)),
                   n(intended_to_meet(sga(5),'hirer protection',
                                              'Commercial Law')))),
       [ModAt3,[['hirer protection'/'consumer protection')],_]],
       [[interpretation_theory('analogia legis')],⟨leg_set_1⟩],_)
```

Now tertiary rules and facts are to be applied to assess the content of the proposed secondary sentence. As on the secondary level, these rules must first be proposed by the user and, for facts, their content must in addition be assessed by the theory T_4 at the quaternary level.

The last goal resolves with `demo` clause `[MP]` resulting as above in four subgoals. The second subgoal of `[MP]` is

```
propose_sent(t(3),
             (demo(n(t(2)),
                   n(intended_to_meet(sga(5),
                                      'hirer protection',
                                      'Commercial Law'))
                  )<=Body_3),
             ⟨mod_3⟩,
             ⟨leg_set_3⟩).
```

where

⟨*mod_3*⟩ is `[ModAt3,⟨mod_2⟩]` and
⟨*leg_set_3*⟩ is `[LegSetAt3,⟨leg_set_2⟩]`.

The adequacy of the secondary rule proposal may be assessed, e.g., with respect to protected interests in the current field. Suppose this is the user's choice and the following `meaningful_sent` is retrieved

```
meaningful_sent(t(3),RuleProp3,Mod3,LegSet3,Text):-
  RuleProp3 =
    (demo(n(t(2)),n(RuleProp2))<=
       adequate(n(RuleProp2),n(ModAt2),LegalField)),
```

```
  Mod3 = [_,[ModAt2,_]],
  LegSet3 =
    [['On protected interests in particular legal fields'|_],
     LegSet2],
  LegSet2 = [LegSetAt2,[[_,_,legal_field(LegalField)],_]],
  Text = '"A secondary rule proposal is legally valid (i.e.,
          belongs to the theory t2 of valid secondary rules)
          if its inclusion accords with the tertiary rules
          for adequacy."

          (If you think assessment of adequacy is inappropriate
          for your case you may reject this rule whereupon other
          possible tertiary rules will be suggested.)',
  accurate_typing(t(3),RuleProp3,[],[],Text).
```

The second argument of `propose_sent` is bound to the tertiary rule

```
(demo(n(t(2)),n(⟨rule_prop_2⟩))<=
   adequate(n(⟨rule_prop_2⟩),n(⟨modat_2⟩),'Commercial Law')),
```

where ⟨*rule_prop_2*⟩ is the `intended_to_meet` sentence.

The third subgoal of `[MP]` calls `[MP]` recursively to prove that the consequence of the proposed tertiary rule follows by proving its body. For each inference step, tertiary rules must be proposed. Say the second `propose_sent` subgoal of `[MP]` returns with its second argument bound to

```
(adequate(n(⟨rule_prop_2⟩),
          n(['hirer protection'/'consumer protection')]),
          'Commercial Law')<=
   assumed_protectionworthy(n('hirer protection'/
                              'consumer protection'),
                           'Commercial Law'))
```

The third subgoal of `[MP]` calls `[MP]` recursively

```
prover(demo(n(t(3)),
            n(assumed_protectionworthy(n('hirer protection'/
                                          'consumer protection'),
                                       'Commercial Law'))),
       [ModAt3,⟨mod_2⟩]
       [['On protected interests in particular legal fields'],
        ⟨leg_set_2⟩],_)
```

and the second `propose_sent` subgoal of `[MP]`

```
propose_sent(
   t(3),
   (assumed_protectionworthy(n('hirer protection'/
```

```
                               'consumer protection'),
                             'Commercial Law')<=NextBody3),
   [ModAt3,⟨mod_2⟩],
   [['On protected interests in particular legal fields'],
    ⟨leg_set_2⟩])
```

results in the binding

```
NextBody3 = reasonable_to_equalize(Reason,
                                   n('hirer protection'/
                                     'consumer protection'),
                                   'Commercial Law').
```

The third subgoal of `[MP]` is

```
prover(demo(n(t(3)),
            n(reasonable_to_equalize(Reason,
                                     n('hirer protection'/
                                       'consumer protection'),
                                     'Commercial Law'))),
       [ModAt3,⟨mod_2⟩],
       [['On protected interests in particular legal fields'],
        ⟨leg_set_2⟩],_).
```

Suppose an unconditional sentence (a fact) is proposed. This satisfies the first subgoal of `[UP]` which is

```
propose_sent(t(3),
   reasonable_to_equalize(Reason,
                          n('hirer protection'/
                            'consumer protection'),
                          'Commercial Law'),
   [ModAt3,⟨mod_2⟩],
   [['On protected interests in particular legal fields'],
    ⟨leg_set_2⟩]),
```

Say, the subgoal succeeds with the bindings

```
ModAt3 = ['actors with similar economic circumstances'/
          'actors with similar financial standing')]
Reason = 'actors with similar economic circumstances'.
```

i.e., in the text below of the ‘reasonable to equalize’ rule

```
'In commercial law it is reasonable to equalize
actors with similar financial standing. '
```

the concept `Reason` is exemplified by `'actors with similar financial standing'`, which has now been substituted for the slightly different notion of `'actors with similar economic circumstances'`.

According to our reasoning scheme, a proposed fact's content must be assessed at the higher adjacent level. The second, third and fourth subgoal of [UP] increases the level index by one, checks for groundness and controls permissibility preparatory to the fifth upward reflection subgoal

```
prover(demo(n(t(4)),
            n(demo(n(t(3)),
                   n(reasonable_to_equalize(
                       'actors with similar economic circumstances',
                       n('consumer protection'/
                         'hirer protection'),
                       'Commercial Law'))))),
       [ModAt4,[['actors with similar economic circumstances'/
                 'actors with similar financial standing'],
                ⟨mod_2⟩]],
       [LegSetAt4,
        [['On protected interests in particular legal fields'],
         ⟨leg_set_2⟩]],_).
```

At the quaternary level, in theory T_4, no rules exist for assessing the proposed lower level 'reasonable to equalize' rule, so this goal resolves with the prover clause [TOP]

```
prover(demo(n(t(J)),n(demo(n(t(I)),n(RulePropI)))),
       ModJ,LegSetJ,ProofJ):-
  J >= 2,
% \+ propose_sent(t(J),(demo(n(t(I)),n(RulePropI))<=BodyJ),
%                   ModJ,LegSetJ),
  external_confirmation(t(I),RulePropI,ModJ,LegSetJ),
  ProofJ = externally_confirmed(sentence_of(theory(I),RulePropI))
```

and the user may or may not accept the content of the 'reasonable to equalize' rule.

Provided the rule is accepted this completes the computation of the fourth conjunct in the antecedent of the *analogia legis* rule with these argument bindings (where ⟨*proof_2*⟩ stands for the proof term)

```
prover(demo(n(t(2)),n(intended_to_meet(sga(5),
                                       'hirer protection',
                                       'Commercial Law'))),
       ⟨mod_2⟩,
       [[interpretation_theory('analogia legis')],⟨leg_set_1⟩],
       ⟨proof_2⟩) ,
```

The following three conjuncts in the antecedent of the *analogia legis* rule are computed likewise which completes the computation of the initial query, repeated here,

```
prover(demo(n(t(1)),n(RuleProp1)),Mod1,LegSet1,Proof1).
```

with these argument bindings (where ⟨*proof_1*⟩ stands for the proof term)

```
prover(demo(n(t(1)),
            n((legal_cons(pay,hirer,letter,goods,price)<=
                 and(actor_1(hirer,goods),
                 and(actor_2(letter,goods),
                 and(unsettled_price(goods),
                 and(demands(letter,price),
                 reasonable(price,goods)))))))),
       [[hirer/vendee,letter/vendor],unspecified],
       [[provision_no(sga(5)),
         provision_category('Determination of Purchase-Money'),
         legal_field('Commercial Law')],unspecified],
       ⟨proof_1⟩).
```

A conclusion is not considered final until the line of arguments leading up to it has been considered and accepted by the user. To this end the user needs a comprehensible presentation of the proof term ⟨*proof_1*⟩. This is the subject of Ch. 7.

CHAPTER 5

'QUERY THE USER' FACILITIES

Being a union of formalized and external knowledge, a semiformal system presupposes an appropriate user interaction. This chapter points out and analyses the aspects we consider central.

OBJECTIVES

As has been illustrated above, the open-textured character of legal knowledge entails that computerized legal reasoning must rely on the ability of the user both for supply and assessment of data. Moreover, user interaction has to be both structured and frequent. An appropriate 'query the user facility' (cf. Sergot [66]) is necessary and should, amongst other things, promote

- meaningful user queries and answers,
- a construction of adequate terms for describing proofs,
- an intelligible explanation of derived conclusions by appropriate display of proof terms, and
- a natural order in which the system poses questions to the user.

Satisfying the first aspect is important. For example, complex query formulas should be composed only of component formulas which are meaningful together [66]. Moreover, the bindings of the individual variables of a query formula should belong to acceptable ranges. The system SIMPLE [66] deals with some of these aspects. In SIMPLE the second and third aspects are promoted by allowing less interesting relations to be declared 'built-in' so that individual steps in their execution are not traced. Also the fourth aspect is dealt with cursorily. If a relation, which is partially defined by program clauses, is declared as 'Ask-about' an extra argument may be added to indicate whether the program or the user should be consulted first.

The frequency of the user interaction distinguishes our case from SIMPLE. It is vital that the questions are posed in an intuitively intelligible order in our system but the approaches mentioned are too coarse for our purpose. In the present study we have delved into all four of the above aspects and below we deal with each in turn.

MEANINGFUL USER QUERIES AND ANSWERS

Let us begin with the question of meaningful user queries and answers. In contrast to the SIMPLE system we do not presuppose that a user by himself is capable of composing an adequate query. This cannot be expected in the legal domain. In our system, user queries are instead composed interactively in order to promote that they will be meaningful. For example, we have typing guidelines which determine the range of acceptable concepts in queries. Also, the various levels in our system have their own strategies for accessing and using the query information data-base for that level. Sergot concludes his study by sketching a so-called 'knowledge base dictionary' with similar possibilities.

ADEQUATE TERMS FOR DESCRIBING PROOFS

The need for adequate proof terms is met as follows. Our system has a 'knowledge driven' inference strategy, i.e., resolving rules are not simply retrieved from the rule base. Both the searching for appropriate rule schemata and the specialization of their content are influenced by the user with the support of encoded type checking knowledge. This demarcates the range of possible rules and may improve efficiency, especially for large rule bases. When the proof term is constructed, only rules that have been deemed relevant are included.

Upward reflection must be included in our system, since it resembles quite closely a central legal inference. SIMPLE's query evaluator is a metaprogram implemented on top of micro-PROLOG

$$\begin{aligned}&\mathit{Evaluate}(\mathit{query})\\&\leftarrow \mathit{Demo}(\mathit{GlobalDB} \cup \mathit{User}, \mathit{query}, \mathit{result}) \wedge \mathit{ExtractOutput}(\mathit{result}).\end{aligned}$$

The *result* parameter is the proof of *query*. The program executor of micro-PROLOG does not make available the proof's structure. Therefore reflection cannot be used in SIMPLE, at least not without losing that part of the *result* parameter which records the inferences the program executor carries out between downward and upward reflection. In our system consisting of $\mathcal{MT}$ and $\mathcal{OT}$ we simulate, in our semiformal metalanguage, upward reflection between a theory `t(J)`—representing T_j of $\mathcal{IT}$—of a formal metalanguage (the language of level j in $\mathcal{OT}$) and an adjacent theory `t(I)`—representing T_i of $\mathcal{IT}$— of its formal object language (the language of level i in $\mathcal{OT}$). Since the semiformal metalanguage takes the whole hierarchy of formal languages as its object language, proof structures may be construed for reasoning between these levels as well as within a certain level and we can use reflection without losing the proof structure. Thus simulated, downward reflection does not improve efficiency, but legal knowledge does not allow downward reflection anyway.

EXPLANATION OF DERIVED CONCLUSIONS

Explanation of derived conclusions raises several problems, for instance, that if the explanation facility displays every resolution step in the proof, the result often becomes too full of details to be really useful. This problem can be reduced by selectiveness either when composing the proof term, or when displaying it, or both. SIMPLE uses the first of these two approaches. As mentioned above, some relations are declared as 'built-in' with the effect that the individual steps in their execution are not included in the proof term. Sergot calls for a neater scheme for displaying proofs. We combine both approaches. Our knowledge driven inference strategy imposes selectiveness when the proof term is constructed by supporting that relevant schemata are retrieved and specialized with meaningful content; this is illustrated below in Ch. 6. The proof term is presented by a piecemeal unfolding of portions appropriate for display and well demarcated parts of the proof may be skipped if the user so wishes; this is illustrated in Ch. 7.

It is awkward to display a proof which demonstrates that a query could not be proven since all failure branches in the computation are included. 'Why not' questions are desirable though. In our system the user is told explicitly that a certain method for legal reasoning, say *analogia legis*, cannot be used to resolve the case, and is asked whether he wishes to try another method. Once all possible methods are exhausted he can be told that no other methods exist, whereupon his query fails.

INTELLIGIBLE ORDER OF QUESTIONS

It is important that questions are posed in an intuitively intelligible order. Our metalogic approach affords a possibility to structure such a session. Metalevel reasoning may answer questions about what are the best orders for posing the questions of the various levels in the object system. In contrast, influencing the order of questions by flagging predicates leaves the inference strategy in control. In a metalogic approach a strategy for generating questions can be designed, which is independent of the inference strategy of the object system. We exploit the metalogic structure of our system to separate the question generator from the formal object level system. The target aimed at is that questions should appear in an order which resembles an average lawyer's lines of reasoning. We have reason to believe that it is appropriate to begin by asking about primary rules at the lowest level and then gradually move up in the hierarchy. Replacing this protocol by some other is simplified by the separation of the question generator from the object system.

DECLARATIVE KNOWLEDGE ASSIMILATION

Sergot proposes an approach for making input and output declarative. With reference to the query evaluator described above, he says "The way to understand declaratively what is going on here is to realize that the combined database, the union between *GlobalDB* and *User*, remains static and fixed. Only the boundary between the two components moves, and then only to make the machine take some burdens of the user." This is paralleled in our system. A session is initiated by the query at p. 51 which should be understood as 'is it in accordance with the higher adjacent level of theory T_1 in $\mathcal{IT}$ to assume that the specified sentence is included in T_1?' The theory T_1 is to be understood as the set of all sentences which satisfy the constraints imposed by the higher levels. All sentences the user may successfully input belong to this set.

In Ch. 6 we illustrate the 'query the user' facilities our system provides by presenting a session. Explanation of derived conclusions is the topic of Ch. 7.

CHAPTER 6

A SESSION

Relying on the user for finding appropriate sentence schemata for the current case and proposing meaningful content for these, our system must provide an adequate support. This chapter presents a session illustrating how the system assists the user in these matters.

Describing a session with our system may give a good practical illustration of the aspect of knowledge assimilation by user interaction and also indirectly of knowledge processing. Since we want to know whether a certain primary rule is included in the theory T_1, each session begins with the same completely non-ground query on p. 51 about the content of theory T_1. As illustrated above the content of T_1 and other theories is fragmentarily described in clauses encoding knowledge about meaningful sentences, i.e., the `meaningful_sent` clauses. Thus the goal is to assimilate new knowledge into a fragmentarily described theory. The assimilation must satisfy certain conditions and the resulting theory should offer a reasonable suggestion for solving the current case. Knowledge to be assimilated into the theory of some level i is proposed by the user and assessed and accepted by rules of a theory at level $i + 1$.

The first task for the system is to identify the relevant part of the legal system for the case the user has in mind. This is accomplished interactively using standard interface and data retrieval techniques, but shown below in a compressed form. This part of the session appears as

```
Which of these legal fields encompasses your case?
        Commercial Law  (cl)    Real Estate Law  (rel)
        Penal Law       (pel)   Procedural Law   (prl)
```
cl. (user answers in italics)

```
Which of these legal problems corresponds to your case?
        Cancellation    (canc)  Completion      (compl)
        Risk            (r)     Delay           (d)
        Determination of Purchase-Money (dpm)
```
dpm.

```
Which of these provisions seems relevant?
        sga(5) If a sale of goods has been made but no price
               settled then the vendee should pay what the
```

```
                vendor demands if reasonable
        sga(6) If ... etc.
```

sga(5).

Now a possibly relevant provision, here the Sect. 5 of the Sale of Goods Act, has been identified. The next task is to determine, first whether it is possible to adapt it to the current case, and second whether the adapted provision may be assimilated into the theory T_1.

The following part of the session settles interactively the proposed modifications and whether these belong to the same type as the concept they replace. The last is to promote that the resulting rule is at least meaningful (obviously we ought not to replace 'vendee' by 'Fido', etc.).

```
This text describes schematically a rule which comprises some
open textured legal concepts.

  If a sale of goods has been made but no
  price settled then the vendee[1] should pay
  what the vendor demands if reasonable.

In this text the open textured legal concepts are underlined.
These may be replaced by similar concepts, i.e., concepts of the
same type, resulting in a modified but perhaps still legally
relevant rule that may be applied to resolve the current case.

The concept: "vendee"
is the rule's example of a concept belonging to
the type:  "actor"
Please modify it: hirer.
The concept:  "vendor"
is the rule's example of a concept belonging to
the type:  "actor"
Please modify it: letter.
```

Suppose the adapted provision is not already known to be included in the theory T_1. Then it may nevertheless be assumed included, provided its inclusion accords with the legal interpretation principles for reducing *lacunae* on level 2, such as, e.g., *analogia legis*. The next question is

[1] Generally, there is a need for a representation of texts which is more structured and comprehensive than underlining to show which are the open textured legal concepts. In our internal representation of the text above, the act of 'a sale of goods' is understood as a certain action involving goods, not necessarily a sale action, carried out by two actors; it is the type of the actors that decides the type of action. That is to say, 'vendee' and 'vendor' decide a 'sale' of goods, and 'hirer' and 'letter' decide a 'hire' of goods.

```
A textual reading of available legal sources does not indicate any
solution to your case. However, various accepted methods exist for
interpreting legal sources and their possible adaptations. Apply-
ing such a method might result in a proposal for a solution.

Three methods are
- "analogia legis" (al),
- "e contrario" (ec), and
- "extensive" (e)
interpretation.

Specify your choice:
```
al.

Analogia legis has been chosen and the relevant rules involved in reasoning by analogy are presented, and the user may either accept or reject them. The first rule encodes the relation between primary rules and the secondary rule *analogia legis.*

```
This text describes schematically a rule

  A primary rule proposal is legally valid (i.e., belongs
  to the theory t1 of valid primary rules) if its inclusion
  accords with the secondary rule for analogia legis.

(If you think analogia legis is inappropriate for your case you
may reject this rule whereupon other possible secondary rules
will be suggested.)

This rule description does not contain any open textured legal
concepts which at this point can be further particularized.
Wherever such concepts occur in its premises they will be treated
instead as the rules matching these premises are activated.

Do you accept the rule?
```
yes.

This answer verifies that the secondary rule for analogia legis should be attempted. The rule is presented:

`This text describes schematically` ..., etc.

```
  A certain rule may be applied to a case not subsumed,
  or at least not with certainty subsumed, under the rule's
  linguistic wording, if the case is not the object of a
  particular explicit rule, if the case has a substantial
  similarity to those the rule is intended for, if interests
  of some importance, which the rule is intended to meet,
  support such an application, and if no contrary interests
  exist recommending the rejection of such an application.
```

`This rule does not` ..., etc.

`Do you accept the rule?` *yes.*

A consequence of this user acceptance is that the system eventually assumes the completely specialized rule of *analogia legis* as a non-logical axiom. At this point in the session the premises of *analogia legis* are still non-ground. Now each premise of the *analogia legis* rule is tried in turn. The conditions vary from field to field under which *analogia legis* may be applied. Therefore, when facts are eventually activated by the rule's premises or the premises of matching rules, the content of these facts must be assessed with respect to the current legal field and other pertinent aspects.

Let us for example consider the 'intended to meet' premise and a fact that may satisfy this premise. The fact is presented as

```
This text describes schematically a fact which comprises some
open textured legal concepts.

  The provision Sect. 5 Sale of Goods Act is to be interpreted as
  if it were intended to protect consumers and similar groups.

The open concepts are underlined.
```

The user is invited to adapt the fact to fit his case.

```
The concept: "consumer protection"
is the rule's example of a concept belonging to
the type:  "protection of weaker party"
Please modify it:
```
hirer protection.

Now the user has proposed a content for this fact which is meaningful in the sense that the modified concepts belong to the same type as those they replace. The content, however, must also be accepted by the rules that lay down the realm of analogical (and other) reasoning in the respective legal fields. These rules appear on the tertiary level, i.e., level 3.

```
You have chosen to try a certain method of inference of legal
reasoning for analysing your case. Such a method must be properly
understood, in particular with respect to aspects pertinent to the
current legal field. The chosen method and its content can be
assessed from various perspectives, e.g., those listed below.

- Interests known or likely to be considered protection-
  worthy in the current legal field (pi)
- General principles of the current legal field
  (e.g., 'legal security' in penal law) (gp)
- The current legal field de lege ferenda
  (i.e., foreseeable future adjustments of the
  conception of justice in the field) (fl)

You may choose any of these alternatives:
```
pi.

Interests considered protectionworthy in the current legal field influence the adequacy assessment of the content of secondary rule proposals. Pertinent tertiary rules are activated in a fashion similar to that described above for the secondary level. First the rule is which encodes the relation between secondary rule proposals and tertiary rules for adequacy.

```
This text describes schematically a rule

  A secondary rule proposal is legally valid (i.e., belongs
  to the theory t2 of valid secondary rules) if its inclusion
  accords with the tertiary rules for adequacy

(If you think assessment of adequacy is inappropriate for your
case you may reject this rule whereupon other possible tertiary
rules will be suggested.)

This rule description does not contain any open textured legal
concepts which at this point can be further particularized.
Wherever such concepts occur in its premises they will be treated
instead as the rules matching these premises are activated.
```

`Do you accept the rule?` *yes.*

`This text` ..., etc.

```
  A secondary rule encoding a legal inference method has an
  adequate content if the interests it maintains are considered
  protectionworthy in the current legal field.
```

`This rule` ..., etc.

`Do you accept the rule?` *yes.*

`This text` ..., etc.

```
  An interest, which resembles some other interest known
  to be considered protectionworthy in a certain legal
  field, may also be assumed protectionworthy, if, in that
  particular legal field, reason exists for equalizing
  the two interests.
```

`This rule` ..., etc.

`Do you accept the rule?` *yes.*

Next a fact is activated and presented as

```
This text describes schematically a fact which comprises some
open textured legal concepts.

  In commercial law it is reasonable to equalize
  actors with similar financial standing.

The open concepts are underlined.
```

As was the case at the lower levels, this allows the user to modify the expression appropriately and, on the adjacent higher level, reasoning is initiated to assess whether these modifications may be accepted.

```
The concept: "actors with similar financial standing"
is the rule's example of a concept belonging to
the type:  "adequate affinities between actors"
Please modify it: actors with similar economic circumstances.
```

For this example, no applicable rules exist on the quaternary level, i.e., level 4, so the user is asked whether he accepts or rejects the tertiary rule.

```
The content of the following text, with adapted open textured
legal concepts,

  In commercial law it is reasonable to equalize
  actors with similar economic circumstances.

has, through external communication, been proposed
for inclusion in the theory on the 3rd level. No
no rules exist, however, on the 4th level for eval-
uating whether the rule's content is adequate.
Therefore it is up to the user to accept or reject
the rule.

Is the content acceptable? yes.
```

This completes the session.

Above, the user's answers have been either in the affirmative or in the negative. To expect such answers might seem too optimistic sometimes, especially when the sample provisions contain vague legal concepts. However, it is not expected that the user be left without help. Connected with most vague concepts there is a set of precedent legal cases which are examples of their application. User interaction systems for proper presentation of precedent cases is a subject dealt with in our previous studies [3,4,5,30]. Coupling our present system with such a system results in a combination similar to that advocated by Bench-Capon [6] between regulation-based and case-based systems. As regards this matter we agree with Wolstenholme's [78] position that precedent cases should, at least for the present, be used merely to help suggest answers and not be represented as the logical definiens of vague concepts.

CHAPTER 7

SEMIFORMALITY AND PROOF VERIFICATION

Instead of restricting attention to its separate parts, it is preferable that the whole line of argument leading to a conclusion is assessed. In this chapter we outline how a proof term—i.e., a record of proof steps—may be exploited to interactively accomplish this.

7.1. PROOF DISPLAY AND USER CONFIRMATION

$\mathcal{MT}$ is a semiformal metalogic theory. In general, user interaction is necessary when a sentence is established as a theorem of $\mathcal{MT}$. There is no way to guarantee the veracity of such an $\mathcal{MT}$ sentence. But this is only as it should be, since the sentence says that a certain expression is a legal rule and no procedure can exist for guaranteeing that such a statement is truthful. However, that $\mathcal{MT}$ statements are reasonable should be promoted as far as is feasible and in a semiformal context the complementary resource available is external assessment. Not only should the sentences proposed as legal rules be assessed, but also the proof of their inclusion in a theory of valid legal rules. Thus, semiformality should be coupled with proof verification.

In Sect. 4.3 we went through a computation of a query during which a term was composed representing the proof of the inclusion of the proposed sentence in the bottom-most theory. Now this proof term should be used to provide the final validation of the proposed sentence.

Using a popular term, the proof of a proposed sentence should not only be reductively approved of, but also holistically. That is to say, ideally, not only should the segments of the proof term be assessed, but also the completed proof as a whole. This poses an obvious difficulty for a human being who cannot survey and grasp more than segments of a long and complicated proof. 'Surveyability' can be enhanced, though, by introducing symbols as abbreviations of complex components of the proof. Depending on which components are replaced, this gives alternative levels of abstractions for displaying the proof term. Ideally, a vast range of possibilities should be provided. Realizing a metaprogram reasoning with the completed proof and composing various levels of abstractions is a matter for future research. For the present, we have only taken the first step to realize such a facility in a metaprogram which replaces components of the proof exceeding a certain size. We give an example below.

7.2. A PROOF DISPLAY

After completion of a session the conclusion is explained by a stepwise unfolding of the proof of the initial query. Below a part of this explanation is described. Since the intention is to illustrate the internal representation, the explanation is described on a low level and certain useful information is excluded from the proof term, such as the textual representation of sentences, what modifications have been made to them and in what legal setting. Facilities which are more appropriate for practical systems may be developed in hypertext and the like.

Let us now illustrate how a proof may be entrusted to the user's acceptance or rejection.

```
sentence_of(theory(1),
            (legal_cons(pay,hirer,letter,goods,price)<=
               and(actor_1(hirer,goods),
               and(actor_2(letter,goods),
               and(unsettled_price(goods),
               and(demands(letter,price),
               reasonable(price,goods)))))))
```

```
holds because
```

```
proof_of(theory(2),
         proved(theory(1),
                (legal_cons(pay,hirer,letter,goods,price)<=
                   and(actor_1(hirer,goods),
                   and(actor_2(letter,goods),
                   and(unsettled_price(goods),
                   and(demands(letter,price),
                   reasonable(price,goods))))))),
         <proof>)
```

`was shown. Do you want "<proof>" further explained?` *yes.*

This says that the proposed primary rule is a sentence of the theory T_1 since there is proof (named `<proof>`) from the theory T_2 establishing that the rule is to be assumed included in theory T_1. The user affirmed further unfolding of the proof. The next steps describe the secondary rules involved. The first encodes the relation between primary and secondary rules.

```
sentence_of(theory(2),
            demo(n(t(1)),
                 n((legal_cons(pay,hirer,letter,goods,price)<=
                      and(actor_1(hirer,goods),
                      and(actor_2(letter,goods),
                      and(unsettled_price(goods),
                      and(demands(letter,price),
                      reasonable(price,goods)))))))))
```

```
abbreviated "sentence_of(theory(2),<goal>)",
holds because

and(rule_of(theory(2),(<goal><=<body>)),
    proof_of(theory(2),<body>,<proof_body>))
```

was shown. Do you want <proof_body> further explained? *yes.*

Here <body> corresponds to the head of the *analogia legis* rule, so next is the actual secondary rule for *analogia legis*:

```
sentence_of(theory(2),
   analogia_legis(n((legal_cons(pay,hirer,letter,goods,price)<=
                       and(actor_1(hirer,goods),
                       and(actor_2(letter,goods),
                       and(unsettled_price(goods),
                       and(demands(letter,price),
                       reasonable(price,goods))))))),
                  n([hirer/vendee,letter/vendor]),
                  [[provision_no(sga(5)),
                    provision_category(
                       'Determination of Purchase-Money'),
                    legal_field('Commercial Law')],unspecified]
                 )
           )

abbreviated "sentence_of(theory(2),<goal>)",
holds because

and(rule_of(theory(2),(<goal><=<body>)),
    proof_of(theory(2),<body>,<proof_body>))
```

was shown. Do you want <proof_body> further explained? *yes.*

Next follows the proof (<proof_body>) of the antecedent (<body>) of the *analogia legis* rule. When a formula is too long to be printed directly it is described abstractly; <body> is a conjunction and is described as such between its first conjunct and the conjunction (<conjunct>) of the remaining conjuncts.

```
sentence_of(theory(2),
   and(not
       casuistical_
       interpretation('Commercial Law',
                    n((legal_cons(pay,hirer,letter,
                                  goods,price)<=
                          and(actor_1(hirer,goods),
                          and(actor_2(letter,goods),
                          and(unsettled_price(goods),
```

```
                              and(demands(letter,price),
                              reasonable(price,goods)))))))),
       <conjunct>)
   )
```

abbreviated "sentence_of(theory(2),and(<g_1>,<conjunct>))",
holds because

```
and(proof_of(theory(2),<g_1>,<proof_g_1>),
    proof_of(theory(2),<conjunct>,<proof_conjunct>))
```

was shown. Do you want <proof_g_1> further explained? *no.*

In <body> the first formula (and on request its proof) is explicitly presented and corresponds to the first premise of the *analogia legis* rule. The second formula, denoted <conjunct>, stands for the remaining part of the antecedent, i.e., the other six premises of the *analogia legis* rule.

Now in turn the respective proofs of each conjunct are explained in <conjunct>, i.e., the remaining premises of the *analogia legis* rule.

Do you want <proof_conjunct> further explained? *yes.*

etc.

Let us take a look at the proof of the fourth premise.

```
sentence_of(theory(2),
            and(intended_to_meet(sga(5),
                                 'hirer protection',
                                 'Commercial Law'),
            <conjunct>))
```

abbreviated "sentence_of(theory(2),and(<g_1>,<conjunct>))",
holds because

```
and(proof_of(theory(2),<g_1>,<proof_g_1>),
    proof_of(theory(2),<conjunct>,<proof_conjunct>))
```

was shown. Do you want <proof_g_1> further explained? *yes.*

The 'intended to meet' sentence is includable as a secondary rule (or fact) of the theory T_2 only if its inclusion accords with the tertiary rules of the theory T_3. So the proof segment <proof_g_1> explains how tertiary rules were applied to infer the acceptance of this inclusion.

```
sentence_of(theory(2),intended_to_meet(sga(5),
                                       'hirer protection',
                                       'Commercial Law'))
```

holds because

```
proof_of(theory(3),
```

```
         proved(theory(2),intended_to_meet(sga(5),
                                           'hirer protection',
                                           'Commercial Law')),
         <proof>)
```

`was shown. Do you want "<proof>" further explained?` *yes.*

The relation between secondary rule proposed for inclusion in the theory T_2 and tertiary rules in theory T_3 is encoded in the first tertiary rule which is presented analogously to the corresponding relation presented above between T_1 and T_2. The explanation of the following tertiary rule is:

```
sentence_of(theory(3),
   adequate(n(intended_to_meet(sga(5),
                               'hirer protection',
                               'Commercial Law')),
            n(['hirer protection'/'consumer protection']),
            'Commercial Law')
   )
```

```
abbreviated "sentence_of(theory(3),<goal>)",
holds because
```

```
and(rule_of(theory(3),(<goal><=<body>)),
    proof_of(theory(3),<body>,<proof_body>))
```

`was shown. Do you want <proof_body> further explained?` *yes.*

We skip the presentation of the following proof steps of application of tertiary rules until the tertiary fact ‘reasonable to equalize’ is reached. This fact is assumed to be included as a sentence of the theory T_3 if its inclusion accords with the quaternary rules of the theory T_4.

```
sentence_of(theory(3),
            reasonable_to_equalize(
               'actors with similar economic circumstances',
               n('hirer protection/consumer protection'),
                 'Commercial Law')))
```

```
holds because
```

```
proof_of(theory(4),
         proved(theory(3),
                reasonable_to_equalize(
                   'actors with similar economic circumstances',
                   n('hirer protection'/
                     'consumer protection'),
                   'Commercial Law')),
         externally_confirmed(
```

```
sentence_of(theory(3),
   reasonable_to_equalize(
      'actors with similar economic circumstances',
      n('hirer protection'/
        'consumer protection'),
      'Commercial Law'))))
```

In our system, no applicable quaternary rules have been encoded, so the proof on the quaternary level consists of a user confirmation of the meaningfulness and acceptance of the proposed tertiary rule.

This completes the proof of the fourth premise of the *analogia legis* rule. The proofs of the remaining three premises of this rule are unfolded and presented likewise.

CHAPTER 8

COPING WITH CHANGE

It is desirable that the formalization of the informal legal theory is both adaptable and structure-preserving. This chapter describes how this is strived for in our system.

8.1. MODIFICATIONS AND STRUCTURE PRESERVATION

Legal knowledge is frequently modified. Therefore a program should be able to *cope with changes* in the knowledge it formalizes. At the same time, the program should be *structure-preserving* ('isomorphic') with respect to this knowledge, cf. Sergot *et al.* [65].

8.2. ADAPTABILITY AND MODULARITY

Bratley *et al.* [12] claim there is a conflict between structure preservation and coping with change. Coping with changes requires modifying "implicit or explicit rules which do not correspond directly to paragraphs in the text of law". It is hard, though, to see exactly what is the conflict. 'Paragraphs in the text of law' may have a structure-preserving representation modulo the text of law, while the mentioned 'implicit or explicit rules' may have the same modulo whatever they originate from. For instance, descriptions of metarules of legal interpretation in legal doctrine give informal counterparts whose structure may be preserved in the formal representation of these rules. Furthermore, even if for some reason it proves impossible to preserve the structure of rules of this kind, this need not impair structure preservation of statutory rules. And, with a metalogic approach we need not even mix structure-preserving and non-structure-preserving representation.

Our metalogic program $\mathcal{MT}$ is a structure-preserving formalization of legal knowledge coping with change. The schemata give a modular, direct and easily altered description of statutory rules and (meta...)metarules of legal interpretation. $\mathcal{MT}$ is structure-preserving both horizontally and vertically, entailing that adjustments can be made locally to the schemata for the (higher level) rules of legal interpretation as well as to the schemata for the ordinary (low level) statutory rules. The level of the knowledge is identified and the appropriate adjustment made to its rule schemata, which then control the computation of accepted rules to be assumed included in theories of the lower adjacent level.

Moreover, since $\mathcal{MT}$ takes as its object language the whole n-level language of $\mathcal{OT}$, we can encode in the formal part of $\mathcal{MT}$, rules coping with global changes which cannot be localized to rule schemata of a certain level of $\mathcal{OT}$. Furthermore, if the legal system has undergone an even more radical revision, a large part of our system will nevertheless remain intact (or, as Gärdenfors [22] perhaps would have put it, at least have a high degree of 'entrenchment') since the structure of well-established principles such as *analogia legis* will hardly be affected. The structure-preserving model of the British Nationality Act [65] is according to Kowalski and Sergot [52] "of limited practical value" since it expresses a "layman's reading of the provision" but in our $\mathcal{MT}$, expert knowledge may be incorporated, e.g., for verifying the correctness of $\mathcal{OT}$, modifying and augmenting it, and for suggesting promising ways to apply its rules.

Bratley *et al.* [12] write "The challenge is to find ways ... so that only a part of the knowledge base has to be reconstructed; better still, we might hope to design an expert system which can suggest ways to react when given information about necessary changes." We have primarily the first objective in mind when claiming that our representation easily copes with change. Its structure facilitates the programmer's maintenance of the represented knowledge in that all adjustments can be made to the schemata separately encoded in the `meaningful_sent` clauses in the metalogic program forming part of $\mathcal{MT}$. Since the adjusted schemata control the computation of lower level rules we have also taken one step towards the second, more ambitious objective of making the 'expert system' react when given information about change.

CHAPTER 9

RELATED WORK

Interpretation assistance for coping with problems of multiple interpretation of provisions has been discussed by Allen and Saxon [1]. In passing, they mention the problem of alternative semantic interpretations but focus on structural interpretation, i.e., the incorporation of various understandings of structural components, such as 'if', 'not', 'provided that', e.g., those arising from altering the choice of component taken as the main connective of a sentence. No analysis is presented of the logical relationship between theories comprising interpretative knowledge and interpreted theories.

Proof terms are objects of discourse in our system. This is because in legal reasoning an activity of indisputable importance is the compilation of persuasive lines of argument pro or contra different and often contradictory legal decisions, and equally important is the assessment of these lines of argument. Proof terms and their utilization have also been discussed by Bench-Capon and Sergot [7] in an outline of a "rule based representation of open texture in law". They note that open texture is a property that holds not only for the components of provisions, the legal concepts, but also for the notion itself, of being a persuasive argument or line of argument. Therefore the proof term should not only be displayed for user communication but also reasoned about. Moreover, they foresee the problem of an infinite regression, since the representation of the 'persuasiveness' of arguments is itself open textured. In contrast to our studies [29,31,32,34], however, no formalization is given or sketched to represent these aspects of legal reasoning. Nor it is analysed in any depth how these aspects are sorted out in informal legal reasoning or the kind of knowledge involved in that process, i.e., no detailed theory is proposed such as our analysis of (meta...)metarules of legal interpretation and their place and function in the multilayered structure of legal knowledge.

Our approach for the use of metaprogramming in logic emphasizes knowledge representation. The fact that the knowledge in the problem domain belongs conceptually to different levels has been reflected as closely as possible. Sterling [68] has investigated another approach for using metaprogramming in logic for multilevel problem solving in which different tasks are allotted to the different levels: planning is carried out at the metametalevel, methods are applied at the metalevel, etc.

Metaprogramming in logic for knowledge representation has been investigated by Bowen [10]. He shows that most traditional approaches to knowledge representation, viz., frames, semantic nets, scripts, message passing, etc., can be cleanly reproduced in logic metaprogramming.

Metaprogramming in logic was touched upon in Sergot's project on representation of statute law in logic programming [65] (the British Nationality Act). Sergot's group presents solutions to some problems and outlines how representation of more advanced legal knowledge could be approached, e.g., deeming provisions and counterfactuals, a thread later picked up, elaborated and to some extent realized by Routen [63], and negation which has later been followed up by Kowalski [51].

In the field of computerized legal knowledge a variety of other projects have been carried out, some of which have reported success in the representation of various aspects of legal reasoning. The weakness of a single interpretation approach is widely recognized and several authors [1,52,67] have emphasized the need to incorporate, in a knowledge system, multiple interpretation of legal knowledge due to its vague, ambiguous, open-textured, fragmentary, schematic or whatever character. When this is written in the spring of 1992, the most up-to-date list of computer programs representing the law we are aware of is a survey compiled by Sergot [67] who categorizes these programs into those embodying one fixed interpretation of legal sources and those striving to go beyond this restriction. He also distinguishes procedural representation from declarative. Our representation strives to qualify for the intersection between the second and fourth of these categories, i.e., a declarative representation embodying multiple interpretations of legal sources. To the best of our knowledge, our research is the first to present a solution in declarative logic programming to the formalizing problems caused by the multilayered and fragmentary character of legal knowledge.

There exist some other projects, however, with certain distant affinities. One is the work by Nitta *et al.* [56] who presented a system in which legal metaknowledge for control is incorporated but where legal metaknowledge for interpreting provisions is treated extra-systematically by selecting the "opinion which is supported by many legal scholars". Moreover, McCarty's TAXMAN II [54,55] includes a structure called 'prototypes-plus-deformations' that seems to have certain affinities to our exploitation of rule schemata for theory construction. It is too vague a notion for a rigorous comparison with our approach, however, and a full implementation of the TAXMAN II theory has never been attempted [55].

In $\mathcal{MT}$, reflection in $\mathcal{OT}$ is defined to occur when the formalization of the informal theory at the current level in $\mathcal{IT}$ does not contain any formula resolving with the current goal. That is, reflection is prompted by insufficiency in the object level theory, not by explicitly calling the reflection rule. This corresponds to implicit reflection, which was proposed by Costantini and Lanzarone [17] and introduced in their Reflective Prolog.

CHAPTER 10

CONCLUSIONS AND FURTHER WORK

Experiences acquired from the implementation of $\mathcal{MT}$ are summarized and directions for future research pointed out.

Five problems connected with the representation of multilayered fragmentary knowledge were enumerated in our previous study [34] and repeated on p. 8 in this thesis. The first three problems were studied in the previous study, i.e., the problems of representing schematic descriptions, firstly of rules, secondly of legal cases serving as interpretation data for such rule descriptions, and thirdly of legal cases from which rules are induced. These three problems could be studied in an idealized setting because the fifth problem, viz., the supply and assimilation of external knowledge, was intentionally disregarded. By removing such idealizing assumptions in subsequent studies [33,35,36] the earlier formalization was turned into a realistic implementation of multilayered fragmentary knowledge. The supply of external knowledge was investigated, in contrast to simply assuming, as before, sufficient knowledge to be available at the various levels in the hierarchy.

We have gained methodological insight concerning the formalization of multilayered, imprecise knowledge. In the study the problem of formalizing imprecise theories is contrasted with the problem of formalizing exact theories. Adapting the system to reality has, besides its practical consequences, motivated a new theoretical analysis of this formalization problem, splitting it into three distinct theories: the informal theory to be formalized, the formal theory, and the informal metatheory discussing the two former and their relationship. These three theories correspond, respectively, to Horovitz' unattainable perfect legal system, its ideal and equally unattainable formalization, and the partial metalogic Horn clause formalization of the available part of this ideal. Our informal metatheory is a semiformal theory whose formal and informal parts, respectively, consist of Horn-clauses and of user interpreted sentences.

Representation of imprecise knowledge raises high demands on user interaction. Appropriate 'query the user facilities' are necessary so as to promote meaningful user answers and queries, to construct and intelligibly display proof terms explaining derived conclusions, and to make the system pose its questions in a natural order.

Coping with change is considered important and problematic for representation of legal knowledge. Our formalization facilitates changes. Firstly, a modular and direct (or 'isomorphic') description of statutory rules and (meta...)meta-rules of legal interpretation is given in our schemata, which can easily be changed, even by the user at computation time. Secondly, the schemata are not rules used directly for resolving legal cases. They serve only as sources from which case-specific rules are generated, at computation time, for the case in question. Thus, the whole rule base is adapted to the case in question.

Several open questions remain for further work and among these we will focus on the following.

- We have provided for multiple semantic interpretation of the concepts in sentences by allowing the user to fill schemata with meaningful content. Thus, the user proposes legal concepts which he believes refer to his factual situation whereupon the system accepts or rejects the thus proposed rule. So far, the user has not been allowed to adapt the logical structure of schemata, changing their structural interpretation by altering the scope of connectives or adding premises. As hinted above, this should raise no fundamental obstacles and may be included as soon as rules of acceptance for such alterations have been established.
- Our system allows queries about conditional sentences whose consequents and antecedents respectively correspond to the consequences and premises of proposed legal rules. Here the problem is whether or not the proposed rule may be considered part of the legal system. Assume however that a contract exists. Then the query must be conditional also with respect to this contract, provided it is valid. A contract is legally valid if its rules accord with the legal system. If the proposed rule follows from the rules of the contract and these follow from the legal system, the contract is in some sense dispensable since the proposed rule would follow directly from the legal system. That is to say, in the perfect legal system there is a rule identical to each enforceable contractual rule. Such rules are distinguished from others however inasmuch as their antecedents have at least one premise relying on the existence of the contract in question, so the contract is still clearly relevant. Apart from these few initial remarks the question is open concerning how queries may be qualified with respect to contracts in our system.
- The problem of inducing schemata from precedent cases was briefly touched upon in our previous study but has so far not been subjected to any thorough analysis. From a theoretical point of view this is a difficult and intriguing challenge. It is important also in a practical perspective since there are totally case-based legal fields and most of the others are case-based to some extent at least. However, with the exception of the specific and difficult problem of inducing a schema from precedent cases, we hypothesize that our representation framework needs only minor adaptations to deal with the problem of case-based reasoning.

So far our multilayered formalization approach has only been applied to knowledge interpretable in the indicative mood. That indicative sentences play an important role in law is claimed in Part II. This position is shared however with several other semantic categories. Formalization of some of these seems to raise problems metalogic programming could cope with, e.g., counterfactuals and deeming provisions. Also, a multilayered metalogic hierarchy is one possible approach for representing iterated modalities. These are matters for further research.

PART II

THE MULTILAYERED STRUCTURE OF LEGAL KNOWLEDGE

Part I of this thesis treats the problems of formalizing a certain description of legal knowledge and reasoning: the multilayered informal theory $\mathcal{IT}$ in which schematic descriptions of legal interpretation rules play the central role. The investigation is devoted to computing science and confined therefore to the problems of representing $\mathcal{IT}$ as a formal theory $\mathcal{OT}$ which is adequate modulo $\mathcal{IT}$ and formally tractable. In chapter 2 it is claimed that $\mathcal{IT}$ is adequate modulo actual legal knowledge, but consonant with the purpose of Part I the argument supporting this claim is presented in a condensed form.

An initial phase of the representation problem is to acquire an understanding of the knowledge domain involved. Our conception of legal knowledge emanates partly from practical experiences with lawyers, and partly from legal philosophy.

Part II is intended to strengthen the claim concerning the adequacy of $\mathcal{IT}$, i.e., to provide an argument that our conception of the multilayered hierarchy of legal interpretation principles is well-founded, that the higher-level principles are important and play a central role, and that they are available only in a schematic form. After introducing the principles, the argument is composed according to the following plan. In order to circumscribe the role of higher-level interpretation principles in legal reasoning, we first attempt to establish the circumstances under which they apply. It will later appear that this presupposes that we acquire some understanding of the semantic problems raised by open textured legal concepts. And inversely, a clarification of how higher-level interpretation principles function will provide an important key to how to deal with these semantic problems. Eventually, we will reach the legal principles composing the fragment of $\mathcal{IT}$ formalized and implemented in Part I, i.e., those involved in the example about *analogia legis* in commercial law.

Part I of this thesis is purely theoretical. In contrast, Part II must be in some sense empirical since we claim that our informal theory has a certain adequacy modulo actual legal knowledge. Our object of study—legal knowledge—is an object of mental activity that does not exist in an objective reality. Therefore, the empirical adequacy of our $\mathcal{IT}$ can depend only upon its correspondence to conceptions of law among lawyers. We can only aspire to establish that $\mathcal{IT}$

is consonant with some more general agreement among lawyers as to how law should be conceived. Legal philosophy can be regarded as a record of such agreements and will be the main source in which we search for a support for our $\mathcal{IT}$. Occasionally, we also refer to experiences acquired in interaction with legal practitioners.

CHAPTER 11

INTRODUCTION

Legal knowledge is inseparably connected with a general problem of interpretation. In this chapter we outline this problem for the particular kind of legal knowledge studied in this thesis. Beginning at the very outset, first introducing concrete and well-known components of law such as statutory rules and other provisions, we gradually approach a fuller sketch of a multilayered hierarchy incorporating higher-level principles of legal interpretation.

11.1. BACKGROUND

When hearing the word 'law', the layman probably equates it with acts and the like. The legal term for this source of law is statute law. Besides statute law we have however several important legal sources, e.g., case-law, equity, legal doctrine, etc. For this thesis the object of study has been statute law and legal sources systematized in a way akin to statute law, e.g., regulations, ordinances, decrees, conventions, treaties, etc. A certain property of these has probably been a dominating source for optimism concerning knowledge representation of law. They are drafted in the form of provisions, e.g.,

(1) The employer's dismissal must have fair grounds.

Although these provisions are often, as here, written as unconditional statements, they can, with one exception, be reformulated to 'if ... then ...' statements, e.g.,

(2) if fair ground for dismissal exists then the employee may be dismissed,

where the circumstances in the antecedent (here 'fair ground for dismissal') lead to the legal consequence in the consequent (here 'the employee may be dismissed').

The exception is when the provision is a so-called 'legislative manifest'. These are expressions of political rhetoric and moral conceptions. For example

(3) Man and wife live under the obligation of mutual faithfulness and assistance.

Legislative manifests are rare—in fact manifest 3 for a long period of time was quite unique in the Swedish legislation. In legal doctrine ([70], pp. 242 ff.) it has been claimed that these exceptions have little effect on legal usage; they cannot be reformulated in conformity with the 'if ... then ...' scheme and are therefore *lege imperfectae*, i.e., law without sanction. Therefore, it is not far-fetched to assume that

(4) only provisions that, without loss of legal adequacy, can be reformulated to 'if ... then ...' statements are of interest to be reproduced in a formalisation.

But this is not unconditionally true. Legislative manifests play a part in that they affect the interpretation of provisions ([70], p. 243). It may be true that their influence in this respect is small. It is however important to realize the significance of the things that exercise influence on the interpretation of legal rules.

The impression that the 'if ... then ...' statements inherent in provisions[1] convey the essence of the legislation is appealing with regard to knowledge representation. However, the impression is incorrect. The statutes belonging to any one branch of law give only an incomplete picture of the legal knowledge of that field. J. C. Gray once expressed this as: "Statutes are sources of law ... not part of the Law itself" ([26], p. 276). Gray's point is that statutory rules are not rules to be applied directly by the courts—they have no independent interpretation: "their meaning is declared by the courts, and *it is with the meaning declared by the courts and no other meaning that they are imposed upon the community as Law.*" ([27], p. 170) (emphasis in original). It cannot be denied that the application of provisions occasionally seems rather mechanical ([39], p. 12). Also in these cases, however, the court's interpretation of the provisions exercises an influence that simply cannot be ignored—in fact, even the conclusion that a provision should be applied mechanically is the result of interpretation. In many cases only a fraction of the factual situations to which the rules apply (or may possibly apply) would be encompassed if direct depictions of the provisions were the sole content of the representation. A knowledge system with such a basis only would be very limited and in many cases not practical. Instead, what needs to be captured is a more abstract kind of knowledge which we may call *legal norms*.

The notion of a legal norm is important but far too complex and disputed within legal philosophy to be investigated in depth here. A rough and incomplete description may suffice however as an indication of the problem. No general agreement exists concerning the norm notion. Different attitudes are embraced by natural law and legal positivism which are two major branches of legal philosophy. Natural law assigns to the norm notion a wider extension than legal

[1] Below the term 'provision' is used interchangeably for the provision as such and the 'if ... then ...' form it may be reformulated to. What meaning is intended will be clear from context.

positivism. Few, if any, contemporary adherents of legal positivism would however equate the norm with the provision. Norms have been described variously but there are some typical opinions in legal doctrine. According to Kelsen "the general norms created by the legislative body are called 'statutes' ..." but "the function of ascertaining the existence of the general norm to be applied by the court implies the important function of interpreting this norm, of determining its meaning" ([46], pp. 257, 143). In a somewhat different context he describes norms as "the expression of the idea that something ought to occur" ([46], p. 36). Other writers have described norms as the meaning of the tokens physically expressing the norm ([59], p. 9), as rule complexes and conceptions which are only incompletely reflected by the provisions ([70], p. 178), as the essence as contrasted with the existence (i.e., statutory rules and the like) of law [45], etc.

Suffice it to say here that legal reasoning is the application of norms, not of provisions. Reaching the norm behind a provision requires that it be interpreted 'legally'. Several things affect this interpretation. To mention a few: the legal branch (penal law, contract law, etc.) to which the provision belongs; legislative manifests such as manifest 3 above; other provisions in the same regulation, also perhaps in other regulations, that ought to be juxtaposed and considered together with the rule at hand; contemporary ethical conceptions, etc.

Thus, the legal interpretation is complex. There do exist, however, general principles for how to perform legal reasoning and these exercise control over the interpretation. Since legal hermeneutics is an important branch of jurisprudence, some of these principles are fairly well understood and documented in legal doctrine. Most principles, however, require further analysis.

It is appealing to somehow preserve the provisions in a representation of legal knowledge, since this enhances transparency. But as we may conclude from the above, logical inferences from such a representation are insufficient for reflecting legal reasoning. If this approach is chosen, the provisions ought as far as possible to be interpreted as what they are, i.e., depictions of legal norms. Providing such an interpretation involves complicated considerations. The first problem we arrive at is

(5) to what extent may principles for how to interpret provisions be formalized and represented?

The representation problem 5 depends on how the principles are understood, which is connected with a recurrent theme in jurisprudence: the question 'What is Law?' In modern legal theory a two-level model is, among others, advanced as the explanation. The model includes *inter alia* how legal rules are interpreted. Its originator, Hart, stresses the important role of what he terms secondary rules (metarules) and the following quotations from his quite influential work 'The Concept of Law' ([39], pp. 92, 79) are enlightening:

> ... they may all be said to be on a different level from the primary rules, for they are all about such rules; in the sense that while primary

> rules are concerned with the actions that individuals must or must not do, these secondary rules are all concerned with the primary rules themselves. They specify the ways in which the primary rules may be conclusively ascertained, introduced, eliminated, varied, and the fact of their violation conclusively determined

and

> ... introduce new rules of the primary type, extinguish or modify old ones, or in various ways determine their incidence or control their operations.

Hart argues that this model of explanation is nothing less than the very essence of jurisprudence ([39], p. 79)

> ... we shall make a general claim that in the combination of these two types of rule there lies what Austin wrongly claimed to have found in the notion of coercive orders, namely, 'the key to the science of jurisprudence'.

Principles for how to interpret provisions have an important role. Deductive interpretation suffices for trivial cases only. Hart enumerates various ways in which secondary rules affect the interpretation. Provisions may be changed into covering more or fewer cases than the meaning of their premises originally suggested, they may be labelled obsolete, new rules may be introduced, etc.

Secondary rules control, e.g., how existing provisions may be interpreted by analogy. This means that a provision is modified in a certain way, making inferable conclusions that originally were not deducible. The extent according to which provisions may be modified depends on the current branch of law. For example, in penal law, legal analogy is subjected to severe constraints but in laws of contract there are fewer restrictions, on condition that the parties are equal. Concerning the more detailed application of analogy, knowledge is thus demarcated to branches of law. The content and form of secondary rules for the inference depend on this and on the case in issue. In spite of these differences it is nevertheless possible to give a schematic description of legal analogical reasoning ([70], p. 71). This schema describes the common characteristics for secondary rules for analogy in all branches of law. It is impossible to settle once and for all what are the content and form of a certain field-specific secondary rule. This holds for analogy as well as for other secondary rules. Therefore, schemata are particularly important. They constitute the most firm knowledge about the rules at these levels. We propose that schemata of this kind and branch-specific knowledge for their interpretation must both be reproduced in an adequate representation of legal knowledge. Furthermore, that this calls for rules at a 'tertiary' level (metametalevel) and even higher levels with respect to provisions. Hart indicates the existence of rules above the secondary level ([39], p. 123)

> Canons of 'interpretation' cannot eliminate, though they can diminish, these uncertainties; for these canons are themselves general rules for the use of language, and make use of general terms which themselves require interpretation. They cannot, any more than other rules, provide their own interpretation.

Extending Hart's terminology, we call such rules tertiary, quaternary, etc. Below we refer to rules above the primary level as 'higher-level rules'.[2]

Our representation should be based on written legal rules, since this improves transparency. However, the *prima facie* interpretation rendered by their written form is not what we strive to capture. Instead we focus on how legal experts conceive legal rules; i.e., the conceptions we here call legal norms. We must keep this perspective before our eyes when we turn to the next topic of study, that of semantics: How do legal experts conceive legal rules and what semantics accords best with this? The questions involved here are of fundamental import for the representation of legal knowledge. Also, they are important for understanding the role of the higher-level rules.

A look at provision 1 reveals several semantic problems. First, the rule contains a deontic expression and seems to be an imperative statement rather than a sentence in the indicative mood. This forces us to ask whether

(6) premises of provisions are capable of truth or if other semantic notions apply instead?

If the answer is that they are capable of truth, there still exists a problem. Rule 1 has the premise 'fair ground'. This is a vague legal concept and the question is

(7) how do truth values get assigned to the legal concepts in a provision?

The existence of vague legal concepts entails that not even the truth notion itself can be taken for granted without analysis. We have the truth theoretical problem

(8) how shall we understand the notion of a premise being true in the legal domain?

[2] That higher-level rules interpret and modify lower level rules distinguishes Hart's theory from the related norm hierachy proposed by Kelsen ([48], p. 221 ff). Kelsen outlines instead the multilayering as a consequence of the existence in a legal order of procedures regulating how other norms are created. In a national legal order, the constitution represents the highest level.

11.2. METHODOLOGY

Below we discuss the semantic problems 6, 7 and 8 and problem 5 about higher-level rules. It will be shown that in order to establish when higher-level rules should be used we must first acquire some understanding of the semantic problems; and inversely, a clarification of how higher-level rules function will give an important key with which to deal with these semantic problems. The four problems 5, 6, 7 and 8 are mutually dependent and we therefore study them together.

Some problems above may raise metaphysical questions, e.g., the semantics of vague legal concepts. Since we focus on how legal experts conceive provisions we can disregard such questions, however. Our objective is to acquire an understanding of and possibly represent legal knowledge. Even if lawyers comprehend the reality in a way that seems incorrect we still aim to represent their knowledge and therefore we do not have to bother about the ontology of their knowledge. The important thing is that we really reproduce how lawyers and jurists comprehend law. We should be able to motivate the view of legal knowledge we advance, e.g., by pointing to legal philosophy and/or empirical experiences from interviews with lawyers, etc.

11.3. DEMARCATION

Reformulated provisions are appealing as a basis for the representation of legal knowledge. In this case however, the representation must support an interpretation of the provisions as norms. Providing an interpretation of this kind involves many problems. In this part of the thesis, the specific purpose is to investigate two categories of these problems, namely

(9) the semantic problems raised by premises and consequents of provisions,

and

(10) the problems of understanding higher-level rules for legal reasoning and the problems of integrating these in a representation.

It will be shown that categories 9 and 10 are interdependent. Before we approach how higher-level rules look, we attempt to establish the circumstances under which they apply. We begin therefore with the semantic problems in category 9. Next, concerning category 10 the emphasis will be put on how the higher-level rules should be understood informally. Here the main object of study will be the use of analogy in legal reasoning. The integration of higher-level rules in a formal representation has been investigated in Part I of this thesis.

11.4. OUTLINE

The rest of Part II is structured as follows: Ch. 12 examines the semantics of legal concepts and settles an appropriate approximation of legal knowledge. It leads to an identification of the case when higher-level rules should be used. Ch. 13 shows the affinity between the higher-level rules used to deal with vague legal concepts and those used for legal analogy. In Ch. 14 we return to the topic dealt with in Sect. 2.1 in Part I and show, with an example from legal analogy, how rules at various levels function together. Ch. 15 concludes part II, the empirical part of this thesis.

CHAPTER 12

THE SEMANTICS OF LEGAL CONCEPTS

Formal semantics should be consonant with the semantic properties of the informal knowledge. This chapter analyses how classical logic relates to the domain knowledge in this respect.

12.1. THE PROBLEM STATED

Conceiving legal reasoning as a process of deductive application of provisions is inadequate, as has been described above. Deductive application presupposes, at least, that it has first been established whether or not the legal concepts in a provision apply to the current factual situation. This is the major and difficult part of legal reasoning. The main difficulty is that, almost always, provisions are composed of legal concepts whose meaning is more or less obscure. This gives rise to semantic questions. We show in the sequel that the answers to these questions are determinative for when the higher-level rules come into operation.

The semantics of legal concepts is dependent upon the sentences of which they form parts, i.e., the provisions. To provide a setting and demarcation for the discussion, let us look at a sample collection of provisions.

In older Swedish legislation, provisions are often drafted as subjunctive sentences. For instance Sect. 5 of the Sale of Goods Act is expressed as

(11) Has a sale of goods been made but no price settled, [the consequence be that] the vendee *pay* what the vendor demands, where this cannot be deemed unreasonable.[1]

In contemporary legislation the subjunctive mood has been replaced by imperative statements. This is indicated either by the use of the imperative mood—exploited e.g., in penal law—or by deontic expressions such as in e.g., Sect. 3 of the Consumer Sale of Goods Act

(12) Should the vendor not deliver the goods as agreed, and this cannot be blamed on either the vendee or some event for which he bears responsibility, the vendee *may then annul* the transaction ...[2]

[1] In Swedish: "Är köp slutet utan att priset blivit bestämt, erlägge köparen vad säljaren fordrar, där det ej kan anses oskäligt."

Provisions and the legal concepts they consist of give rise to difficult semantic problems. One reason is that provisions contain linguistic constructions whose semantics is not fully understood. Take for example the mood of provision 11 and the modality used in provision 12. The semantics of sentences in the indicative mood is fairly well understood, e.g., by the model theory of classical logic. Subjunctive and imperative statements have a less readily understood semantics, however. These and other things that provisions may contain are difficulties of considerable proportions within the philosophy of logics. To quote Davidson: there remains "a staggering list of difficulties and conundrums" ([18], p. 321), to be solved before we know how to represent adequately subjunctive and imperative sentences and sentences containing probability and causal statements, adverbs, attributive adjectives, verbs of belief, perception, intention, action, etc. Provisions may contain anything in this list.

12.1.1. The Reformulation Problem. We do not intend to get involved in the philosophical problems of analysing the structure of sentences containing linguistic forms such as those just listed, for two reasons. The first is as follows. Though a one-to-one correspondence (a direct mapping) between provisions and their representation is desirable due to its superior intelligibility (or transparency) we cannot allow this to fix the interpretation of the provisions. For example, though we prefer that in the formal representation, say, an imperative statutory rule appears in (or close to) its original linguistic form, the interpretation of the rule may not be fixed or restricted to this linguistic form. As explained above, this study focuses on how lawyers reason about provisions. It would be a mistake to presuppose that a lawyer is bound to the literal form in which a legislative provision is couched. It is quite clear that the lawyer is free to interpret provisions differently and also that he does so. The reason why a provision is given a certain formulation is to be found in the issuer, i.e., the law-maker. It is natural from the law-maker's point of view to express provisions as commands. At the time provision 11 was made law, the common legal drafting technique for expressing commands was to use the subjunctive mood (*coniunctivus hortativus*); the contemporary way is to use the imperative mood or deontic expressions. What is natural from the issuer's point of view does not have to be natural also for the addressees. The group of addressees we are interested in here are lawyers carrying out the practical legal work of applying provisions to factual situations. It is implausible that these are restricted to reasoning about provisions such as rules 11 and 12 as subjunctive and imperative sentences, respectively. Rather, in his practical work the lawyer is free to disregard the linguistic form the law-maker once gave to the provision and apply instead the interpretation suitable for the current function. When this

[2] In Swedish: "Har säljaren ej avlämnat varan i rätt tid och beror det ej av köparen eller händelse för vilken denne står faran, får köparen häva köpet."

function is to settle whether provisions apply to factual situations, the indicative mood seems adequate. Provision 11 would be applied reformulated to indicative sentences which could look something like:

(13) If a sale of goods is made and no price settled and the vendor demands a reasonable amount and the vendee pays that amount then the vendee fulfils his obligation.

and

(14) If a sale of goods is made and no price settled and the vendor demands a reasonable amount and the vendee does not pay that amount then the vendee fails to fulfil his obligation.

and

(15) If the vendee fails to fulfil his obligation then a lawsuit is successful.

etc.

These sentences contain the connective 'if ... then ...'. If the sentences are to be viewed as purely indicative we must understand this connective as a material implication. Thus conceived, the sentences express legally acceptable states of affairs. For instance, sentence 15 expresses that the following state of affairs is not legally acceptable: 'the vendee fails to fulfil his obligation and a lawsuit is unsuccessful'. The lawyer applies provisions to factual situations. A factual situation may be understood as a state of affairs. Interpreting the provisions as describing legally acceptable states of affairs is not indisputable but seems at least fairly natural from the perspective of rule application.

12.1.2. The Problem of Truth and Indicative Sentences. The position put forward above may be questioned by maintaining that in legal reasoning, norms are treated, not as descriptions being indicative sentences, but as imperative sentences, cf. e.g., Ross [61,62]. Within philosophy and legal theory it has been put forward that legal norms (and their parts) are not capable of 'truth': "Many philosophers and logicians have thought that norms are essentially void of truth-value, 'outside the realm of truth and falsehood', belong to 'practical' as distinct from 'theoretical' discourse." [81]. These opinions simply cannot be ignored. It is necessary to give a further and more detailed motivation concerning why indicative sentences and truth are appropriate for depicting the kind of knowledge focused on in this study.

We believe that the question whether or not norms are capable of truth cannot be answered in general, that is independently of the category of lawyers for whom the question is put. Since we claim norms are, not the written legal rules, but the conceptions or interpretation lawyers hold of these rules, we must allow that different lawyers maintain disparate conceptions. The crux of the matter is then what exercises influence over this choice of conception. We believe that the function of the lawyer has an important role. Depending on the function carried out by the lawyers in respective groups the idea that norms cannot be comprehended as true or false will be intuitively acceptable or directly counterintuitive.

Consider for instance a law-maker whose main function is to issue norms. How does this function affect his understanding of these norms? Well, when issuing a norm the law-maker gives a command. In this prescriptive role the norm does not convey information, either to the law-maker or to the addressees, that something is or is not the case. Therefore, one readily assumes that a lawyer who carries out the legislative function does not comprehend norms as true or false.[3] Norms are rather orders, permissions or authorizations to him, cf. e.g., Kelsen ([48], p. 73) "... the norms enacted by the legal authority, imposing obligations and conferring rights upon the legal subjects are neither true nor false, ..."

But again these are the norms as viewed by the law-maker and not by the group of addressees of interest here, i.e., lawyers who in practical legal work apply norms to factual situations. A lawyer carrying out this function does not comprehend norms as commands, only as descriptions to be applied to factual situations. He uses his knowledge of norms in a way that exhibits no real difference from the way he uses his knowledge of the surrounding world. In both cases he applies descriptions—of norms or objects—to factual situations. If a description applies, it is true relative to the factual situation, or else false. In this context, the thought that norms are incapable of truth seems counterintuitive. At least no convincing arguments have been presented so far against representing this type of norm knowledge as indicative sentences and therefore classical logic could very well be suitable. Observe that we do not argue that there is only one particular clear-cut function of the respective categories of lawyers. A judge for instance is in one perspective an addressee of the command inherent in the norm issued by the legislator. But he does not use this interpretation of the norm when he applies it to a case, only when he chooses to obey the order to apply the norm, and these are distinct activities.

When applying a rule, a lawyer's conception of this rule corresponds closest to what Kelsen calls 'rules of law'. "The statements formulated by the science of law ... do not impose obligations nor confer rights upon anybody; they may be true or false" ([48], p. 73). A rule of law states the existence of a norm and describes it, and it is accordingly true or false. Raz gives the following more formal description of rules of law (which he terms normative statements): "One may say that a normative statement has the general form that p ought to be

[3] In legislative material the law-maker sometimes illustrates a norm's applicability by giving model cases. When he assumes this function the law-maker treats the norm as true or false.

the case, and that it is true if, and only if, there is, in a certain normative system, a norm to the effect that p ought to be the case" ([60], p. 47). The 'descriptive' view of legal norms is rather accepted; for discussions cf. Hedenius [40,41], Strömholm ([70], p. 95). Kelsen for a long time defended this view, e.g., "But the 'ought' of the legal rule does not have a prescriptive character, like the 'ought' of the legal norm[4]—its meaning is descriptive. This ambiguity of the word 'ought' is overlooked when ought-statements are identified with imperative statements" ([48], p. 75). Later, Kelsen abandoned this position ([47], p. 2). As explanation he gave the existence of contradictory norms. This, however, is not a convincing argument against classical logic. A theory including a contradiction can never be accepted as the solution to a legal case. It is a completely different matter that of two consistent theories, which are inconsistent together, each can be applied to a case with equal success. But in this case the theories are also mutually exclusive and the choice between them must be made at the metalevel and surely by a consistent metalevel argument.

Studies of practical legal work strengthen the assumption that when lawyers apply norms they conceive these as indicative rather than imperative or subjunctive sentences. They describe their knowledge in terms of indicative sentences and seem to treat them as such in their reasoning [37]. This is not to say that the imperative approach ought to be rejected though. Important groups of lawyers carry out other functions than the application of norms, e.g., law-makers. Judges have besides the function of applying norms an executive function as well. A norm issued by the law-maker conveys a command that the judge should (i) apply the description inherent in the norm to relevant factual situations, and, (ii) in his turn issue a command as to the execution of the consequences of the norm. Perhaps imperative sentences are necessary in a representation of this kind of legal knowledge and possibly also of the layman's knowledge of law.

This was the first reason why a part of Davidson's list [18] might be unnecessary in a representation of legal reasoning. The second is as follows. It cannot be denied that there may exist cases where the representation of the kind of knowledge we are interested in must include linguistic constructions such as those listed by Davidson. For instance, lawyers may conceive provisions as causal statements and consequently the interpretation of 'if ... then ...' as a material implication is inadequate. However, that sentences may be causal statements or comprise other linguistic forms in Davidson's list is a general problem in knowledge representation. Until a more common agreement as to their representation in general is settled, it would be premature to discuss whether some specific

[4] We have chosen in this study to use the term 'legal norm' for the conceptions and rule complexes behind a provision, cf. e.g., Strömholm ([70], p. 178). Kelsen uses the term in a different sense here: a 'legal norm' is the prescription a legal rule expresses to its addressees. Therefore, a legal norm is an imperative sentence, in contrast to the rule of law which describes a legal rule and is an indicative sentence.

analysis is needed when sentences of this kind express legal arguments.

In summary, suggesting that deontic logic is always indispensable since all law can be regarded as normative is an extreme position. With Sergot ([67], p. 63) (and McCarty) we believe

> Deontic modalities are encountered in legislation and the need for reasoning with the deontic concepts does arise in legal analysis and problem solving. But deontic concepts have no extraordinary status in this respect: they are one instance of a whole range of modalities and concepts that deserve attention too.

12.1.3. Vague Legal Concepts. This background renders it reasonable to apply a demarcation to the study here. We shall discuss only the semantics of provisions that lawyers reformulate and interpret as indicative sentences, and, moreover, only those provisions that comprise only the classical truth-functional connectives.

With this demarcation the remaining source of semantic problems is that the legal concepts in the provisions often have an obscure meaning. The problem of subsumption—i.e., to assess whether the legal concepts apply to a certain factual situation—is an essential element of practical legal work. When vague concepts are involved, this problem is particularly awkward.

Recall rule 2:

(2) if fair ground for dismissal exists then the employee may be dismissed.

This rule is applicable to factual situations involving an 'employee', a 'dismissal', and 'fair ground'. Intuitively, one can easily distinguish the first and second from the third. We are inclined to understand the two former as the names for physical or primitive 'facts' (or 'factual situations') but the latter as something 'non-physical' and more diffuse, e.g., a name for a moral judgement. The reason why we deem 'employee' and 'dismissal' as 'primitive' is that, even though borderline cases exist, we have a fairly clear conception of what an 'employee' and a 'dismissal' are. Concerning 'fair ground' we have no such clear conception. For want of a more appropriate name we will call legal concepts 'sharp' when a clear conception exists of what they denote, and 'vague' when not. We take open textured concepts to be a superclass of vague concepts, the former affected by vagueness only in the borderline cases, the latter affected in their core as well. We call both vague concepts since it is vagueness that constitutes the difficulty in both cases.[5]

All legal concepts have what Hart terms an *open texture* ([39], p. 124), i.e., "whichever device, precedent or legislation, is chosen for the communication of

---

[5] There are various reasons why legal concepts are vague. The concept 'fair ground' is an example where the legislator has intentionally omitted guidelines

standards of behaviour, these, however smoothly they work over the great mass of ordinary cases, will, at some point where their application is in question, prove indeterminate". This means that which concepts are to be considered as sharp and vague, respectively, depends upon several factors. Of course, some concepts are intrinsically vague, e.g., 'fair ground'. Also, however, concepts such as 'employee' and 'dismissal' may be sharp under some circumstances but vague under others. The user-category exercises influence here. If unacquainted with the conditions under which someone is to be regarded as an employee, users will consider that concept as vague. The legal matter in hand is also of importance. There is no ultimate and universal level of detail that can be reached, e.g., by defining 'employee' as a 'person' that has the properties so-and-so, cf. Gardner ([23], p. 95), Oliphant [57]. Most often there is some other legal matter where this definition does not suffice. Concerning 'person' we have for instance the law on abortion in which 'person' itself becomes a vague concept: whether a physician who illegally aborts a foetus be sued for manslaughter will depend upon whether the foetus is considered a person or not. Often, the purpose of a knowledge system is determinative for which concepts are to be considered as vague and sharp, respectively. That is, regardless of how vague a legal concept in fact is, it will simply be assumed as sharp if it falls outside the delimitation of the knowledge system. A knowledge system is restricted to conveying knowledge about some open-textured or vague concepts and will support assessment of their applicability, whereas other concepts, perhaps in spite of an obvious relevance, are left out. Such a delimitation is always necessary since a matter of law is in itself open textured and cannot be completely circumscribed.

In a reasonably expressive formal language we have means for depicting entities such as objects, relations, and functions. The set of entities supposed to be depicted by the symbols of a language may be termed the language's conceptualization [24].

A lawyer has a depiction of the physical reality. When the lawyer uses sharp legal concepts in his reasoning he refers to this depiction. For the sake of argument, assume that 'employee' is a sharp legal concept. This would mean that there existed a fairly precise idea of what the statement 'John Doe is an employee' stands for—'employee' denotes a relation and 'John Doe' an object and both the relation and the object are 'intelligible'—and therefore it would be reasonable to depict the concept as the sole primitive fact: '$Employee(John\_Doe)$'. But how is the position concerning vague legal concepts (and regarding the borderline cases of other open-textured concepts)?

---

concerning how the concept should be understood. This is a law-making technique that may be adopted when the legislator considers it appropriate to let legal usage, doctrine, etc., gradually give content to a legal concept. Another reason is that the use of sharp legal concepts would render the law unmanageably huge and hard to survey.

Vague concepts, e.g., 'fair ground' in rule 2, give rise to difficult problems. Since we have no precise idea of what a vague concept refers to, it is not reasonable to depict these solely as primitive facts: e.g., '*Fair_ground*(*Dismissal*, *John_Doe*)'. A lawyer uses vague concepts extensively in his reasoning. It is interesting to examine to what extent such concepts can be represented; this question is crucial for the success of legal knowledge systems. Before the representation problem is undertaken, however, we must acquire a deeper understanding of vague legal concepts, especially of the lawyer's knowledge of these concepts. We will term this topic the *epistemic question* henceforth.

On the one hand we have legal rules and legal concepts that form part of these and on the other hand a counterpart to these concepts—the entities in the conceptualization. Meaning is given to the legal concepts by their relation to the entities of the conceptualization. What semantic notion most adequately reproduces this meaning in a legal context is controversial. For instance, performance values have been proposed as a competitor to truth values [79]. In the preceding we have argued that truth is an appropriate notion—at least for the kind of legal knowledge we focus on in this study. But a conclusion that truth is an appropriate notion does not suffice. We need a deeper understanding of the truth notion as applied to legal concepts. The reason is the existence of vague legal concepts. That a concept is vague means that it is not clear what it refers to. A denotational semantics[6] for such concepts therefore raises several questions. So, specifically for vague legal concepts, we must investigate what it means for them to be true. We will call this the *truth theoretical question* henceforth.

To summarize: Vague legal concepts raise 'epistemic' and semantic questions the latter of which has a truth theoretical nature. The 'epistemic' question is:

- What does the lawyer's knowledge of vague legal concepts consist of?

and the 'truth theoretical' question is:

- What does it mean for a vague legal concept to be true?

12.2. THE EPISTEMIC QUESTION

We begin by considering the epistemic question. Legal knowledge consists to some extent of 'primitive facts' which it is reasonable to depict as objects, relations, and functions. Is it possible to describe the knowledge about a vague legal concept in terms of such facts? If so, to what extent? Are there any other intelligible entities besides these facts?

[6] We think this term is preferable to 'referential semantics' since denotation or designation is a semantic notion (what expressions do) while reference is more a pragmatic one (what speakers do), cf. ([28], p. 70)

The point of departure below is the way expert lawyers prefer to describe their knowledge. The experiences recounted were acquired during a project focusing on the treatment of vague legal concepts in practical legal work within the Swedish legal system [37].

As an example, let us consider the vague legal concept in rule 2 'fair ground' for dismissal. Asked to share with us his knowledge of this concept, an expert lawyer would begin, in all probability, by enumerating a number of descriptions of factual situations in which he knows that fair ground has been deemed applicable and non-applicable, respectively [37]. These descriptions often contain 'precise' objects, relations and functions. A 'positive' factual situation—i.e., a situation in which 'fair ground' applies—is e.g.,

(16) The absence (a function) of the employee (an object) is t and t is greater than (a relation) one week and the position (a relation) of the employee is ordinary and no extenuating circumstances (an object) have been alleged (a relation).

and a negative factual situation is e.g.,

(17) The absence (a function) of the employee (an object) consists of minor late arrivals (a relation) and the employer (an object) has not taken (a relation) any measures in order to deal with the situation (an object).

An expert's knowledge of a vague concept often encompasses many descriptions of factual situations such as 16 and 17. These form two collections, a 'positive' and a 'negative'. The positive collection includes descriptions of factual situations relative to which the concept has been considered applicable, e.g., by a court. The negative collection includes ditto relative to which the concept has been considered non-applicable.

Although the extensions of the collections vary as legal usage evolves and moral attitudes in society change, it is fair to say that they constitute the firmest content of a vague legal concept. A vague concept may lack collections, however. An obvious presupposition is that the applicability of the concept relative to the factual situations has been examined, e.g., by a court in an adjudication, or by the legislator as model situations described in legislative materials, or in legal doctrine by a legal theorist whose authority, on this specific issue of law, is widely accepted in the judiciary (an authority of this kind is rare but nevertheless exists concerning some issues) or in some other way which is recognized as authoritative.

Besides these more or less 'reliable' collections we will encounter other entities. When invited to give a fuller account of the concept, the expert will probably point out general criteria that must always be fulfilled in order for the concept to be applicable. Concerning 'fair ground', one such general criterion is that

(18) the misbehaviour implies the unlikelihood of betterment.

Criteria such as criterion 18 are true and false, respectively, relative to two sets. These are subsets of the positive and negative collections of the vague concept to which the criterion belongs. For instance, the criterion 'unlikelihood of betterment' is clearly the case relative to the description 16 and not the case relative to description 17. Were this its sole content, a general criterion would not add anything to our understanding of the vague concept. But something more than this seems to exist. In most cases, an expert lawyer has some ability to judge whether or not a new situation fulfils the criterion. In making this judgement it seems clear that he makes use of external aspects, such as moral opinions. It is however reasonable to assume that, rather than making a 'free judgement', the lawyer reasons in quite a structured way concerning these aspects (cf. e.g., Sundby's theory of norms ([72], p. 214) and for discussions on this theory Bing [8]). Normally there exists a genuine possibility of making a correct prognosis of the outcome of a legal case ([70], p. 181). That the lawyer reasons in a 'structured way' means that higher-level rules govern his reasoning.

12.2.1. Approximations of Legal Knowledge. In this study we are concerned with certain fundamental aspects of legal knowledge. An abstraction from details of lesser importance is necessary. Therefore, let us from now on understand legal expert knowledge according to the summary given in statements 19 and 20.

(19) The knowledge concerning a sharp legal concept, such as 'employee' in rule 2, consists of one or several primitive facts. Concerning a vague legal concept, such as 'fair ground', the knowledge consists partly of two collections of descriptions of factual situations relative to which the concept has been considered applicable—or non-applicable.

(20) Besides the two collections the knowledge concerning a vague concept also includes general criteria. It is plausible that the test, whether or not a new situation fulfils these criteria, is govern by higher-level rules. These would thus control the classification of new situations to either of the two collections.

Of course, statements 19 and 20 are only approximations and simplifications of actual legal knowledge. But they are sufficiently adequate for discussing the problems taken up in the remainder. Approximations 19 and 20 have support in practical evidence. They correspond to the way expert lawyers prefer to describe their knowledge [37].

Being a resort when approximation 19 does not tell whether a legal concept is applicable or non-applicable, approximation 20 should be used with care and only when necessary. It provides at best a well-founded guess concerning the applicability of the concept. How do approximations 19 and 20 complement each other? We need not resort to the higher-level rules of approximation 20 in the case where the denotations of a concept suffice to establish whether it is applicable or non-applicable. The higher-level rules discussed in approximation 20 are applicable precisely when it is uncertain whether a vague legal concept denotes a certain situation description. Approximation 20 is thus applicable precisely when approximation 19 does not provide any further understanding regarding the applicability of a legal concept.

An adequate denotational semantics for legal concepts should coincide with approximation 19. Such a semantic system should encompass the applicability or non-applicability of a legal concept in all circumstances under which approximation 20 should not be resorted to.

The next section contains a discussion intended to advance towards such a semantic system. It leads to the identification of circumstances under which further discussion of semantics seems unfruitful and therefore a resort is motivated to the higher-level rules of approximation 20. If correct, this identification will shed more light also on the question of whether classical logic accords with legal concepts. Subsequently, problems of understanding these higher-level rules are discussed and the difficulties of capturing them in a representation are touched upon. This latter discussion proceeds independently of whether the mentioned identification is correct.

12.3. THE TRUTH THEORETICAL QUESTION

One objective of this study is to acquire a deeper understanding of the higher-level rules of approximation 20. But it is equally important to establish when the higher-level rules should be resorted to at all. That a sentence contains vague legal concepts does not mean that higher-level rules must always be used to assess whether it applies to a factual situation. The circumstances must be identified when this can be done without the use of higher-level rules. The connection with semantics here is that, ideally, we should choose a semantics such that a truth value is assigned to the sentence in all cases where it is natural to do so without the help of higher-level rules.

We call this a truth theoretical question simply because its answer depends on how the truth notion is conditioned in the legal domain. We have no intention of assigning the legal truth notion to truth theories such as correspondence, coherence theories, etc.

12.3.1. Truth Values of Composite Sentences. A vague legal concept is 'unknown' relative to a situation description that deviates from its two collections. A concept being 'unknown' in this sense may be a component of a sentence, e.g., a provision. Higher-level rules should be used to deal with the truth value 'unknown' but only if the sentence as a whole is 'unknown' relative to a factual situation. That is, higher-level rules should be used only if it cannot be established otherwise that the truth value of the whole sentence is 'true' or 'false'.

The trivial case where higher-level rules should not be used is of course when the sentence does not include any components the truth values of which are ‘unknown’. But also when the sentence does, there exist cases where it would be natural for a lawyer to assign truth values ‘true’ or ‘false’ without the use of higher-level rules.

Take as an example the sentence that describes ‘theft’ in the Swedish legislation. A direct translation of the original provision in the Swedish penal law is: “Anyone who takes, without permission and with the intention of misappropriation, what belongs to another should be sentenced, if the misappropriation entails injury, to at most two years imprisonment for theft”. The sentence is composed of two vague legal concepts: ‘taking without permission’ and ‘intention of misappropriation’. As described above, the provision can roughly be reformulated to the sentence:

(21) *if* ‘taking without permission’ *and* ‘intention of misappropriation’ *then* ‘theft’.

‘Robbery’ is also composed of several legal concepts. A translation of the original provision is: “Anyone who steals by means of violence, or threat constituting, or from the victim’s point of view apparently constituting, a pressing danger or . . . should be sentenced to imprisonment for robbery”. Reformulated this provision becomes:

(22) *if* theft is carried out by: ‘the means of violence’ *or* ‘threat constituting a pressing danger’ *or* ‘threat apparently constituting a pressing danger from the victim’s point of view’ *then* ‘robbery’.

In the antecedent of sentence 21 the first legal concept is ‘taking without permission’. Assume a judge had established that this concept was false relative to the current situation description. Quite indisputably he would then stop his investigation and conclude that the antecedent as a whole was false. And if he failed to establish the truth or falsity of ‘theft by the means of violence’ in the antecedent of sentence 22 his conclusion would still be that the antecedent as a whole is true if ‘threat constituting a pressing danger’ proved to be true.

It is reasonable assuming that a lawyer treats legal concepts that have the truth value ‘unknown’ as described above. That is, despite the presence of ‘unknown’ a disjunction is considered ‘true’ if one of its disjuncts is true and a conjunction ‘false’ if one of its conjuncts is false. The sentence will get the truth value ‘true’ or ‘false’ if it is ‘true’ or ‘false’ independently of the truth value the ‘unknown’ concept later proves to have. So the presence of a concept being ‘unknown’ does not necessarily mean that higher-level rules must be applied. This is necessary only if the sentence as a whole has the truth value ‘unknown’, that is, when the truth values of its components do not suffice to establish the truth value of the whole sentence.

12.3.2. Semantic Systems. We have discussed above how the truth value ‘unknown’ functions in the legal domain. Is this behaviour reflected in any existing semantic system? To approach that question, let us consider first what is the semantic status of the truth value 'unknown' compared with the classical truth values 'true' and 'false'?

Is 'unknown' a third truth value on a par with the classical truth values 'true' and 'false'? If so there could exist an incompatibility between vague legal concepts and classical logic, since classical logic is bivalent. Does bivalence accord with vague legal concepts? In order to answer this question we must first clarify what is meant by 'bivalence'.

Advocating that bivalence means that

(23) each wff is either true or false and these truth values are always known or easily established

would be to rule out of court any truth values besides 'true' and 'false'—our 'unknown' for example—and recognize as bivalent only formalisms such as propositional calculus and the like. This is not how bivalence is commonly understood.

The common understanding of bivalence is that

(24) each wff is either true or false, but not both

without any statement concerning how these truth values are established. Classical predicate calculus is bivalent in this sense and also e.g., Tarski's truth theory [28,73].

Conception 24 does not deny intermediate 'truth values'. A formula may be 'unknown'. 'Unknown' may however not be a truth value on a par with 'true' and 'false'. That is to say, the assignment of 'unknown' may not be understood as representing *precisely* the idea that the formula is neither true nor false. But it may be understood as an epistemic variant of a classical truth value, i.e., that a classical truth value exists for the formula but in this case it is concealed from us.

Does 'unknown' have the same semantic status as 'true' and 'false' or is it just to be considered as an epistemic variant of these? Let us again take a look at practical legal work. What happens when a court (or some other authoritative body) examines the truth value of a concept relative to an arbitrary factual situation? There are three possible results of such an investigation: the court may conclude

(25) that the factual situation shall be classified to the positive collection

or

(26) classified to the negative collection

or

(27) that the claim be rejected, since it is absurd.

Concluding that the concept is 'unknown' relative to the factual situation is no alternative. A prohibition exists against such conclusions, the *non liquet* prohibition, cf. Ch. 13. Result 27 does not express that the court considers the concept as 'unknown' relative to the factual situation. Rather, it says that the concept cannot be meaningfully applied to the factual situation.

At any given point in time the knowledge of a vague concept consists of two collections of factual situations which have previously been adjudicated by the judiciary. What about other factual situations? Relative to these the truth value of the concept will be unknown to us as long as they have not been adjudicated by any authoritative body. So in practical usage 'unknown' seems to be an epistemic notion. 'Unknown' is not a definite truth value. As the investigation progresses it will be replaced by 'true' or 'false'.

It seems thus that 'unknown' in a legal context is just an epistemic variant of the classical truth values and consequently vague legal concepts are bivalent in the sense of conception 24. So the compatibility between legal concepts and classical logic cannot be refuted by reference to the fact that vague legal concepts are 'unknown' relative to factual situations not yet tried.

Some doubts may still remain concerning how bivalence accords with the court's result 27. Take as an example the legal concept 'heir' and a factual situation involving Fido and his offspring Caro. It may be conceived as meaningless to apply 'heir' here. Dogs do not inherit each other. To say that 'Caro is the heir of Fido' lacks a classical truth value relative to the factual situation may seem more adequate than saying that it is false.

This is of no consequence for the applicability of laws of classical logic, however, since the conditions under which something is meaningful are 'preparatory' to logic. If the concept 'heir' is meaningful with respect to human beings only, there exists a logical type 'human being' that must be satisfied before it is at all meaningful to apply the laws of logic to the concept, cf. Russell [64], von Wright [80]. It is thus not meaningful at all to apply the 'law of excluded middle' (bivalence) to the concept 'heir' before it has been affirmatively answered that the objects involved are 'human beings'.[7] Provisions are drafted carefully. Only in exceptional cases, if ever, could doubts arise as to when it is at all meaningful to discuss their applicability. At least it cannot be a far-fetched approximation to

[7] "The idea that one must first know the preconditions for a statement to be meaningful, before one can enquire about the preconditions of its veracity, has with good reason been called one of the greatest and most consequential discoveries of modern logic. One may say that it bridges *logic* and *semantics*." ([80], p. 74). Original text in Swedish: "Tanken att man först måste känna villkoren för att ett uttryck skall vara meningsfullt, innan man kan fråga efter villkoren för dess sanning, har med rätta kallats en av den moderna logikens största och följdrikaste upptäckter. Man kan säga, att den slår en bro mellan *logik* och *semantik*."

assume that nothing of value is lost if meaninglessness is sorted out by the use of logical types.

A system may thus be bivalent even though it contains formulas lacking truth values if the lack is conceived only as an epistemic variant of the classical truth values 'true' and 'false'. Examples of such systems are Kleene's three-valued logic ([49], p. 333) and Kripke's truth theory [53].

In the examples about 'theft' and 'robbery' above, we described how a lawyer in practical work probably treats a sentence containing legal concepts with truth values 'unknown'. Roughly, the semantics is: composite wffs, that have components the truth value of which is 'unknown', will only get the truth value 'true' ('false') if they are 'true' ('false') irrespective of the truth values these components later prove to have. Otherwise the composite wff is 'unknown'. For instance, a disjunction is true if one of its disjuncts is true and a conjunction is false if one of its conjuncts is false. This corresponds to the semantics for this case in Kleene's and Kripke's logics.

The intention here is not to give a definition of the legal truth notion. Before that can be done, many difficult questions must be examined and answered. We will confine ourselves here to the tentative observation that there exist bivalent semantics that seems to correspond rather well to an intuitively acceptable view of sentences containing vague legal concepts. When a composite wff is used to represent a sentence containing vague legal concepts, there seems, for instance, to exist no reason to let a component, the truth value of which is 'unknown', dominate the whole compound as in Bochvar's logic [9].

Summary. We have in this chapter discussed the notion of truth in a legal context. We have concluded that vague legal concepts give rise to a truth value 'unknown'. This truth value is only an epistemic variant of the classical truth values 'true' and 'false'. Sentences containing concepts being 'unknown' are assigned truth values if sufficiently many of their remaining components have truth values similar to the semantics for this case in Kleene's three-valued logic. The question remains: how shall we treat the case when

(28) a sentence, composite or non-composite, has the truth value 'unknown' according to this semantics?

That is, where the truth values of true or false components do not suffice to give the sentence of which they form part the truth value 'true' or 'false', for instance, when one of the conjuncts in a conjunction is 'true' but the other 'unknown'.

A concept has the classical truth value 'true' relative to a factual situation if it is known to denote something in the factual situation, 'false' if it is known not to denote anything in the factual situation. Case 28 holds precisely when the legal concepts that have truth values in a sentence do not suffice to determine the truth value of the whole sentence. Reaching further by discussing denotational semantics does not seem possible here. Thus we have reached the objective of this chapter. The circumstances are identified under which the higher-level rules of approximation 20 should be resorted to. They will coincide with case 28 if the discussion above is correct. Though Kleene's and Kripke's logics have a semantics which seems to some extent adequate, a three-valued logic cannot be accepted as a representation of vague concepts, since in case 28 it leaves a sentence with the truth value 'unknown'. At least a tentative truth value must be proposed for this case.

Here we must modify our previous statement that the semantic system should be such that it includes all cases where higher-level rules should not be resorted to. The semantic system is in itself an interpretation principle, i.e., it is determined by higher-level rules. These higher-level rules should be such that they encompass all situations where higher-level rules that modify the knowledge of lower levels should not be resorted to, i.e., all cases except case 28.

In the next chapter we come nearer to how higher-level rules should informally be understood. We show that the higher-level rules of approximation 20 are similar in function to another category of higher-level rules, those for reasoning by analogy from provisions. The chapter ends with an informal description of a schema for higher-level rules in that category.

CHAPTER 13

VAGUE CONCEPTS AND LEGAL ANALOGY

Reasoning with vague legal concepts has both similarities and differences vis-à-vis analogia legis. This chapter discusses these and presents a proposed schema for secondary rules for analogia legis.

13.1. THE INFLUENCE OF HIGHER-LEVEL RULES

The characteristics of the legal truth notion collected in Sect. 12.3 led up to case 28 in which we could no longer refer to semantics for assigning truth values to legal concepts. Establishing truth values is particularly awkward in this case. We lack a theoretical basis for truth value assignment but are at the same time forced to assign at least a tentative truth value. It is not necessary to motivate in detail why case 28 is important. It is quite obvious that if case 28 is not dealt with somehow, this limitation will render a potential knowledge system almost useless. We will in this chapter discuss some steps towards a treatment of case 28.

However, for the sake of completeness we must first note that not all cases except case 28 are necessarily unproblematic. With its current definition, case 28 includes only sentences whose truth value cannot for the time being be established, since too many of their components are still 'epistemically unavailable' due to lack of empirical data. This excludes the sentences which are possible theorems of logic but so far neither revealed nor disproved as such, e.g., Fermat's last theorem. These are epistemically unavailable, not due to lack of empirical facts but due to the undecidability [13,75] of their possible non-theoremhood. We could extend case 28 to include these as well, but in all practical respects we can probably simply ignore them.

It is rarely the case that a vague legal concept or a sentence containing such a concept is true or false relative to a situation description. In practice, case 28 seems to be the most likely. The truth value of the vague legal concept or the sentence will be 'unknown' and the problem is to decide to which collection the factual situation should properly be assigned. The higher-level rules of approximation 20 come into operation here and must be represented in order to deal with case 28.

Before we can begin to discuss how to actually represent higher-level rules we need a deeper understanding of rules at these levels. We shall try to acquire this by examining reasoning by analogy in law. Legal analogy has been analysed by several writers in legal philosophy. The results of these analyses can be used as a source for finding higher-level rules for legal analogy and therefore constitute an appropriate object of study here. In the sequel a background concerning analogy is given and it is explained why dealing with legal analogy gives rise to difficulties resembling those encountered with the the higher-level rules of approximation 20.

13.2. LEGAL ANALOGICAL REASONING

Inference by analogy is frequently used by human beings and the difference between analogy and deduction was observed early. Aristotle distinguished analogy from deduction and induction ([43], p. 52 f.) and in the work 'On methods of proof', last century B.C., Philodemos pointed out that analogical inferences do not follow from the premises by necessity as deductive ones do ([76], p. 142). In contrast to deduction we have rather elusive intuitions concerning inferences based on similarity, such as analogy. In legal reasoning, such inferences are frequently used. Besides analogy we have e.g., the commonly used inference *e contrario*, i.e., a case not subsumed under a provision may be settled contrary to what the rule prescribes if the 'silence of the law' can be assumed intentional; the candidate provisions are selected due to their similarity to the case in issue. In this light, pessimism concerning the possibility of representing legal knowledge may be near to hand. However, though it cannot be denied that some affinity exists between general and legal analogy, the prospects of representing the latter are far better. Legal analogy is more circumscribed and well-understood than general analogy. There exists a demand for a legal justification and within jurisprudence descriptions of the inference rule have evolved.

During recent centuries, analogy has become an inference of increasing importance to legal reasoning. The reason why may be summarized as follows. In our modern society it is obviously impossible to provide a written rule for every conceivable situation of legal relevance. Yet, a court is always obliged to find a conclusion for every situation brought before it according to the *non liquet* prohibition (except for situations that obviously lack every legal implication imaginable)[1] ([70], p. 375, [71], pp. 146 ff.). The court may not reject a situation on the ground that it is not covered by any regulation. This would be to commit a serious malpractice termed *déni de justice*[2].

Furthermore, *predictability* is a fundamental component in the Swedish and related legal systems. "This requirement follows from the very idea of a legal

[1] cf. above the distinction between meaningless and meaningful.

[2] The original formulation of this principle is "Le juge qui refusera de juger, sous prétexte du silence, de l'obscurité ou l'insuffisance de la loi, pourra être poursuivi comme coupable de déni de justice". It was introduced in Code Civil art. 4 after the French Revolution and gradually acquired acceptance in the Scandinavian legal system during the 19th century.

system. There must exist an identifiable set of norms ... the ordinary member of society must have a possibility to recognize the existence of law regulating his conduct." ([20], p. 125). The principle of predictability is part of the idea of 'legal security', whose closest counterpart is the doctrine of 'rule of law' in the Anglo-American legal systems ([70], pp. 231, 390). (We avoid the more common term 'principle of legality' here since it is sometimes used to denote a variant of the principle of predictability restricted to penal law.)

On the one hand a court must thus adjudicate every factual situation brought before it and on the other hand maintain the principle of predictability. This gives rise to a conflict concerning factual situations not directly subsumed under existing provisions. The compromise solution is to preserve, while obeying the *non liquet* prohibition, as much predictability as possible. Sometimes this is achieved by applying an inference called *analogia legis*.

Analogia legis may be applied to factual situations that bear some resemblance to existing provisions. A presupposition is that the factual situation at hand obviously lies outside the ambit of existing provisions, i.e., analogy may not be confused with the process of extensive interpretation, cf. von Savigny and Hammond ([38], p. 570). Accordingly, in the provisions involved the legal concepts may not be so vague that they allow the factual situation to be subsumed ([70], p. 408). The court strives to choose the provision being, in the relevant sense, most similar to the current factual situation. This provision is modified to a rule which subsumes the factual situation, a rule that becomes part of legal usage and may be applied in later cases. The court thus constructs new analogous rules. In order to maintain predictability and distinguish relevant from irrelevant similarity the court must use higher-level rules when constructing these analogous rules.

Besides factual situations that resemble existing provisions we have those that bear no resemblance whatsoever to provisions in existing regulations. For them the solution must be derived from general legal principles. The process is called *analogia iuris* but, despite its name, it is not an analogical inference according to legal doctrine ([70], p. 411). We shall not discuss that inference further here.

13.3. VAGUE CONCEPTS COMPARED WITH ANALOGIA LEGIS

Analogia legis and analogical reasoning prompted by vague legal concepts exhibit affinities as well as differences. Let us first consider the affinities.

13.3.1. The Affinities. A functional affinity exists between *analogia legis* and the analogical reasoning in the context of vague legal concepts. To illustrate this let us consider what the two inferences do. *Analogia legis* is to construct from a provision such as

(29) If a *sale* of goods has been *made* but no price settled then the *vendee* should pay what the *vendor* demands if reasonable.

an analogous rule that subsumes a new factual situation. For example,

(30) If a *hire* of goods has been *agreed* but no price has been settled then the *hirer* should pay what the *letter* demands if reasonable.

A vague legal concept such as 'fair ground' has a positive and negative collection of factual situations e.g.,

(16) The absence of the employee is t and t is greater than one week and the position of the employee is ordinary and no extenuating circumstances have been alleged.

and

(17) The absence consists of minor late arrivals at work and the employer has not taken any measures in order to deal with the situation.

Here analogical interpretation is used to determine whether a new factual situation is similar enough to the previously adjudicated factual situations and should therefore be included in either of the two collections. Analogical reasoning of this kind as well as the analogical reasoning termed *analogia legis* deal with new factual situations and do similar things. In both it is judged whether there exists enough similarity between a new factual situation and, in the case of *analogia legis*, an antecedent of a provision, or, in the case of analogical reasoning in a context of vague legal concepts, a previously adjudicated factual situation.

13.3.2. The Differences. There are also differences between analogical reasoning in the context of vague legal concepts and *analogia legis*. Not unexpectedly, these are caused by the fact that the two inferences deal with sharp and vague concepts, respectively.

Let us begin with *analogia legis*. Due to the principle of predictability the structural similarity between the factual situation at hand and the antecedent in the rule is important here. Assume that the legal concepts are all sharp in the antecedent of the rule $\mathit{Sale} \wedge A_2 \wedge \cdots \wedge A_n \to B$. We require that the factual situation at hand should be interpreted to $\mathit{Sale} \wedge A_2 \wedge \cdots \wedge A_n$ for the rule to apply directly and to $\mathit{Hire} \wedge A_2 \wedge \cdots \wedge A_n$ for the rule to apply to hire *ex analogia*, etc.

When, on the other hand, vague concepts are involved the structure of the factual situation is less relevant for the outcome. Applying 'fair ground' to a factual situation involving a new form of misdemeanour means that this factual situation is compared for similarity with factual situations in the concept's two collections. But similarity between factual situations does not presuppose any structural affinities, i.e., a likeness or parallelism in relations. Resemblance between two factual situations may be a similarity only in qualities or a similitude between their abstractions. Depending on the kind of similarity involved in each case, the inference may be termed analogy, extensive interpretation or whatever.

Analogy of this kind between a new factual situation and one previously subsumed under a vague concept does not result in the construction of a new rule but adds a new factual situation to one of the collections. This is not an instance of *analogia legis*.

Of course, this is not to say that similarities other than structural are irrelevant for *analogia legis*. They are not, because they influence the final assessment and acceptance of a constructed analogous rule. In the search for candidate rules (schemata) for *analogia legis* the demand for an isomorphism between factual situations and rule antecedents is helpful however. And it is obviously more difficult to understand and formalize similarity based upon abstract qualities only than ditto also based on structure.

Thus, the statement that the higher-level rules of approximation 20 give rise to technical difficulties having a similar character to those of the higher-level rules for *analogia legis* does not extend to say that the higher-level rules are the same in both. The two rule categories have similar functions though. The problem of how they should be understood is similar in the sense that if we understand the higher-level rules for *analogia legis* we are at least closer to understanding the higher-level rules of analogical interpretation between factual situations.

A presupposition for *analogia legis* is that the rules involved contain only concepts being more or less sharp, cf. above. Therefore, it is reasonable to assume, as we shall do, that the extensions of the legal concepts are known and fixed in the rules used as examples below. This implies that we can elaborate our examples at a 'rule level' where the reasoning deals with the words of the rules and the possibility of replacing these with other words having distinct extensions. Also, we shall assume that the extensions of the replacement words are known and fixed.

We describe *analogia legis* in greater detail in the next section.

13.4. A SCHEMA FOR ANALOGIA LEGIS SECONDARY RULES

As planned at the outset a point has now been reached where concrete examples of (schemata for) higher-level rules can be given. The inference rule for *analogia legis* has been described as follows in legal literature

(31) A certain rule may be applied to a case not subsumed, or at least not with certainty subsumed, under the rule's linguistic wording, *if* the case is not the object of a particular explicit rule, *if* the case has a substantial similarity to those the rule is intended for, *if* interests of some importance, which the rule is intended to meet, support such an application, and *if* no contrary interests exist recommending the rejection of such an application. ([70], p. 71)[3]

This is not a secondary rule for *analogia legis*. It is only a schema that, if it is correct, describes what is common to all secondary rules for *analogia legis*. *Analogia legis* is not a generally valid inference rule. Its content is so to say 'domain dependent'. That is, properties of the legal field may restrict or extend the applicability of the inference rule. The inference rule for analogy is, for example, much more restricted in labour law than in contract law. In penal law the restriction is to the extent that it is commonly argued (although perhaps not adequately cf. ([21], p. 125)) that analogy is precluded within the field. It is important to realize the significance of this 'field related' knowledge.

For an actual *analogia legis* inference rule, the applicability is restricted to a certain legal field. An inference rule of this kind deals with the provisions of the legal field in question and is therefore properly understood as a secondary rule. Secondary rules for *analogia legis* must all comply with the schema 31 (which we for the sake of argument assume is the 'correct schema'). In order to actually formulate a secondary rule, schema 31 must be specified by the use of knowledge specific to the current legal field. This knowledge and schema 31 are thus at a metalevel with respect to real secondary rules [31]. Extending Hart's terminology, we may term the field-specific knowledge tertiary rules and schema 31 a tertiary schema . Here it is important to note that it is not possible to specify in advance field-specific secondary rules for *analogia legis*. This must be done dynamically in each case. We will explain why and how in the next chapter.

Summary. In this chapter we showed the affinity between the higher-level rules for treating vague legal concepts and those for *analogia legis*. We introduced the notion of a schema for higher-level rules and cursorily began to identify the levels to which various fragments of legal knowledge belong.

In the next chapter we shall deepen our study of the levels of legal knowledge with an example from *analogia legis*.

---

[3] Original text in Swedish: "en viss regel kan användas på ett fall, som icke eller åtminstone ej med säkerhet omfattas av regelns språkliga ordalydelse, *om* fallet ej är föremål för en egen uttrycklig regel, *om* fallet företer betydande likhet med dem som regeln avser, *om* intressen av viss vikt, som regeln avses tillgodose, talar för en sådan användning och *om* inga motstående intressen talar däremot."

CHAPTER 14

LAYERS OF LEGAL KNOWLEDGE REVISITED

Preparatory to the problems dealt with in Part I, an abridged account was presented in Ch. 2 introducing the informal legal theory $\mathcal{IT}$. In the present chapter this account of $\mathcal{IT}$ is reconnected to, extended and commented on.

To clarify how rules at different levels interact, we now give a concrete example with *analogia legis*. The schemata and rule proposals involved are those of Fig. 2.1 but we repeat them here to facilitate reading and make Part II self-contained.

Consider first the ordinary provision 29 above, from Sect. 5, Swedish Sale of Goods Act. Provision 29 is not applicable to sale of goods only. It could be analogically applied to e.g., hire of goods (cf. above), or extensively interpreted, or interpreted by inversion (*e contrario*), etc. In this sense, provision 29 embraces numerous primary rules among which one and only one is the rule given by a literal reading of the tokens physically expressing the provision. Provision 29 is a schema for all these rules and since it deals with primary rules we call it a secondary schema . Between secondary schemata and primary rules, the relation is given by secondary rules. For example, between schema 29 and real primary rules, the relation is given by secondary rules that could look something like

(32) SGA, Sect. 5, may be applied to a case not subsumed, or at least not with certainty subsumed, under its linguistic wording *if* the case is not the object of a particular explicit rule in any act belonging to commercial law, *if*, according to the present conception of justice in commercial law, the case has a substantial similarity to those Sect. 5 is intended for, *if* such an application is without detriment to consumers, and *if* protection of free competition does not recommend the rejection of such an application

which is just an example of how a secondary rule for *analogia legis* in commercial law could possibly look. In the same way there exist tertiary schemata for secondary rules and tertiary rules that express the relation between these schemata and the secondary rules. Secondary rule 32 originates from tertiary schema 31. Information about the relation between schema 31 and secondary rules such as 32 is given by tertiary rules. We give three examples, two from commercial law

(33) In commercial law *analogia legis* may not be applied in a way that imposes a burden upon the consumer.

and

(34) In commercial law *analogia legis* may not be applied in a way which counteracts free competition.

and one from penal law.

(35) In penal law for *analogia legis* only the following similarities are relevant: 'man' may be substituted for 'woman', etc.

In Ch. 2 the two figures 2.2 and 2.1 give, respectively, an illustration of the levels of legal knowledge and an example of schemata and rules in this hierarchy for *analogia legis*.

The notion of schemata for rules at different levels may seem abstract and hard to grasp. Since they are the sole help a lawyer has when examining a case, the schemata are important, however. In advance, the lawyer never knows what are the actual rules applicable to a certain case. Concerning these rules the only knowledge he has is the schemata on different levels and the task is to extract the applicable rules from these. Of course, since no real rules exist on any level the extraction process involves a moment of discretion, i.e., a point at which the lawyer just decides, without further consideration how a schema should be transformed into a real rule.

Since no real rules exist in advance, lawyers cannot just select a rule and generalize it to a schema. Schemata are specialized to rules top-down, not vice versa. At some arbitrary topmost level a schema is selected and by discretion transformed into a real rule used for specializing schemata to real rules at the level below. In turn, these rules are used in the same way for the next level below, etc. However, top-down rule specialization must be contrasted with bottom-up reasoning. The search for adequate schemata is controlled by data originating from sources such as the descriptions of the case in hand, and therefore supplied by a bottom-up process.

In previous chapters we discussed the expert lawyer's knowledge of vague legal concepts. This knowledge, we argued, consists partly of two collections of factual situations. In this chapter we propose that a lawyer has only a schematic knowledge concerning legal rules. This does not extend with the same force to the factual situations. Concerning these, the lawyer has a more firm knowledge.

Among writers who have made similar observations concerning the schematic nature of legal norms we note Bing ([8] p. 12). In a comment on Sundby's theory of norms Bing observes a category of so-called 'non-deterministic' rules, i.e., rules that cannot be predefined. According to Sundby [72] these rules are used to assess vague legal concepts. Sundby calls them 'assessment rules' and they correspond to our higher-level rules. Bing concludes that, since assessment rules cannot be predefined, Sundby's theory, if it is correct, implies important limitations for the formal representation of legal norms in computerized systems.

It has been shown in Part I of this thesis that the non-feasibility of predefining higher-level rules—or any other legal rules for that matter—raises no fundamental hindrances for their formal representation, on condition that these rules can at least be schematically described. And if even this is impossible, the claim that 'assessment rules' exist seems far-fetched.

In the preceding chapters we offered a fairly detailed account concerning the rules on the lowest level, i.e., the provisions ('primary rules' or 'secondary schemata' as we from now on conceive them, cf. Fig. 2.1). We argued for the possibility of reformulating these rules to 'if ... then ...' sentences; we advanced that the lawyer under certain circumstances conceives these sentences as indicative sentences; we discussed what semantic system would be appropriate for these sentences. With respect to higher-level rules we are not in a position where we can carry out a similar analysis. The reason is that we do not have a deep enough knowledge about these rules. Most often they are not available in the same form as lower-level rules. In the legislation higher-level rules appear only occasionally as provisions. Some higher-level rules are accounted for in legal texts. But the large body of the higher-level rules exist only in the minds or subconscious of the lawyers. It is a very difficult problem to establish what is the structure and content of these rules. General conclusions regarding their form and semantics cannot be drawn at this stage. In this thesis the separate examples we give of higher-level rules all happen to be 'if ... then ...' sentences and we have assumed without further notice that the semantics of classical logic applies to them.

One thing deserves explicit mention. A transformation from a secondary schema to a primary rule is apt to be structure-preserving. At least we can assume this with some certainty by virtue of the principle of predictability. In general however, structure preservation is not a demand for transformations from schemata to higher-level rules. That the structure is more or less preserved in the examples given in Fig. 2.1 is a coincidence.

Summary. Legal knowledge does not consist of rules that can be applied mechanically. Legal reasoning is to establish the form and content of candidates for applicable rules. The knowledge used is schemata for rules. Rules at an even higher level give information about the relation between these schemata and real rules. The lawyer's knowledge of these rules is only schematic as well.

CHAPTER 15

CONCLUSIONS

The multilayered model of legal knowledge is summarized and ideas indicated for its wider application.

Part II of the thesis was intended to strengthen the claim of adequacy of the informal legal theory $\mathcal{IT}$ formalized in Part I. Consonant with the view that the empirical status of a theory of law is to be established modulo a more general consensus among lawyers of what law is, support has been sought mainly in theories of legal philosophy. This has emphasized the important role of legal interpretation for reproducing the legal norms underlying statutory rules and only incompletely reflected by these. Also, Hart's categorization of primary and secondary legal rules supports the claim that legal knowledge is multilayered. Furthermore, we have examined when the principles of legal interpretation apply by investigating the semantics of legal concepts and how they operate by composing a fragment of $\mathcal{IT}$ for *analagia legis* in commercial law.

Part II is the preparatory formalization phase of Part I. We have considered the relation between the formal semantics of our formal language and the semantic questions raised by the properties of the informal knowledge. We have seen that vague legal concepts need not be incompatible with the bivalence of classical logic. Furthermore, our multilayered model is based on the insight that the lawyer's semantic interpretation of legal rules is not fixed. Influence is exercised by higher-level rules but also by the lawyer's current function. Here in particular the norm applying function seems consonant with a semantic interpretation of legal rules as indicative sentences while other functions may call for a different interpretation.

Part II of the thesis indicates even more clearly than Part I the wide range of topics for future research. Though vague legal concepts caused some initial hesitation, the preparatory formalization phase in Part II has been restricted to knowledge with a semantics being at least not *prima facie* incompatible with classical logic. Other less tractable knowledge categories exist. But in fact, a preparatory formalization analysis of some of these may reveal reasons for attempting a multilayered metalogic formalization. For instance, representing iterated modalities in a metalogic hierarchy has been attempted by Priest [58]. Other interesting knowledge constructs with a more discernible metacharacter are counterfactuals and deeming provisions.

Our approach presupposes that schemata for sentences of various levels can be identified. It would be interesting, therefore, to investigate how far our approach can be pursued in a more full-scale implementation.

CONCLUDING REMARKS

Let us now sum up the complete thesis. In retrospect what have we accomplished? Our main objective was to propose a representation framework, with a well-founded theoretical basis, that could cope with features prevalent in realistic knowledge, that is, the sentences composing the informal theory are neither precise nor unequivocal but only known in a schematic form that requires interpretation.

We have developed this representation framework as a semiformal interactive metalogic program, which is well-founded in the sense that it can be interpreted as a first-order theory. The central idea is that this metalogic program should compose, interactively with a user, a proposal for a formalization of the informal knowledge, testing and revising it, until it is satisfactory. The latter is tested for, partially by simulating the computation in this formalization of the formulas proposed as rules of acceptance for lower level formulas, and partially by applying rules of meaningfulness when formulas are proposed. The input of the metalogic program is meaningful rules, specialized from schemata in co-operation with the user, and its output an object theory—a suggested formalization of the informal knowledge—which can also be interpreted as a first order theory.

The strength of our framework has been tested on a complicated informal theory—a description of legal knowledge as a hierarchy of arbitrary many levels of schematic rule descriptions. However, though legal knowledge has been used as an example, nothing is particularly 'legal' with our representation framework and it should be possible applying it in other domains also where the knowledge is fragmentary and schematic. It is worth noting that such knowledge seems to be tractable only if describable canons for its interpretation exist. Such canons make the knowledge multilayered and how this can be coped with has been thoroughly investigated in this thesis.

REFERENCES

1. L. E. Allen, S. S. Saxon, “More IA Needed in AI: Interpretation Assistance for Coping with the Problem of Multiple Structural Interpretations”, *proc. Third Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1991), 53–61.
2. J. Barklund, A. Hamfelt, “Hierarchical Representation of Legal Knowledge with Metaprogramming in Logic”, accepted for publ. in J. Logic Programming, (1991).
3. J. Barklund, A. Hamfelt, S.-Å. Tärnlund, “A Legal Reasoning Assistant”, *proc. 9th Symp. on Legal Data Processing in Europe, 1989*, (Strasbourg: Council of Europe, 1991), 133–4.
4. J. Barklund, A. Hamfelt, J. Wünsche,“A Modular Architecture for Knowledge Systems”, *Computational Intelligence* III, eds. N. Cercone, F. Gardin, G. Valle, (Amsterdam: Elsevier, 1991), 247–62.
5. J. Barklund, A. Hamfelt, J. Wünsche, “Building Modular Legal Expert Systems”, *J. Expert Systems with Applications* vol. 4, no. 4, (New York: Pergamon Press, 1992).
6. T. J. M. Bench-Capon, “Practical Legal Expert Systems: The Relationship Between a Formalisation of Law and Expert Knowledge”, *Computers, Law and AI*, eds. M. Bennun, A. Narayanan, (Ablex, 1989).
7. T. Bench-Capon, M. Sergot, “Toward a Rule-Based Representation of Open Texture in Law”, *Computer Power and Legal Language*, ed. C. Walter, (New York: Quorum Books, 1988), 39–60.
8. J. Bing, “The Problem of Finding a Precedent”, (Oslo: University of Oslo, Norwegian Research Center for Computers and Law, 1989).
9. D. A. Bochvar, “On a three-valued logical calculus and its application to the analysis of contradictions”, *Matematiceskij sbornik* vol. 4, no. 2, (1938): 287–308.
10. K. A. Bowen, “Meta-level Programming and Knowledge Representation”, *New Generation Computing* 3 (1984): 359–83.

11. K. A. Bowen, R. A. Kowalski, "Amalgamating Language and Metalanguage in Logic Programming", *Logic Programming*, eds. K. Clark and S.-Å. Tärnlund, (London: Academic Press, 1982), 153–72.

12. P. Bratley, J. Fremont, E. Mackaay, D. Poulin, "Coping with Change", *proc. Third Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1991), 69–75.

13. A. Church, "A note on the Entscheidungsproblem", *J. Symbolic Logic* vol. 1, (1936): 40–1; correction, *ibid.*: 101–2.

14. K. L. Clark, "Negation as Failure", *Logic and Data Bases*, eds. H. Gallaire and J. Minker, (New York: Plenum Press, 1978), 293–322.

15. A. Colmerauer, H. Kanoui, M. van Caneghem, R. Pasero, P. Roussel, *Un Système de Communication Homme-Machine en Francais*, (Marseille: Univ. Aix-Marseille II, 1973).

16. S. Costantini, "Semantics of a Metalogic Programming Language", *proc. Second Workshop on Metaprogramming in Logic*, ed. M. Bruynooghe, (Leuven: Katholieke Universiteit Leuven, 1990), 3–18.

17. S. Costantini, G. A. Lanzarone, "A Metalogic Programming Language", *proc. Sixth Intl. Conf. on Logic Programming*, eds. G. Levi and M. Martelli, (Cambridge, Mass.: MIT Press, 1989), 218–33.

18. D. Davidson, "Truth and Meaning", *Synthese* 17 (1967): 304–23.

19. S. Feferman, "Transfinite Recursive Progressions of Axiomatic Theories", *J. Symbolic Logic* vol. 27, no. 3, (1962): 259–316.

20. M. Flodin, *On Valid Law and Valid Moral Norms*, (Stockholm: Norstedts, 1986).

21. Å. Frändberg, *Om analog användning av rättsnormer*, (Stockholm: Norstedts, 1973).

22. P. Gärdenfors, "The Dynamics of Normative Systems", *preproc. Third Intl. Congress—Logica, Informatica, Diritto—Expert Systems in Law* vol. I, ed. A. A. Martino, (1989), 293–9.

23. A. Gardner, *An Artificial Intelligence Approach to Legal Reasoning*, (Cambridge, Mass.: MIT Press, 1987).

24. M. R. Genesereth, N. J. Nilsson, *Logical Foundations for Artificial Intelligence*, (Los Altos: Morgan Kaufmann Publ., 1987).

25. F. Giunchiglia, A. Smaill, "Reflection in Constructive and Non-constructive Automated Reasoning", *Meta-Programming in Logic Programming*, eds. H. Abramson and M. H. Rogers, (Cambridge, Mass.: MIT Press, 1989), 123–40.

26. J. C. Gray, *The Nature and Sources of the Law*, (New York: Univ. Press, 1909).

27. J. C. Gray, *The Nature and Sources of the Law*, second ed., revised by R. Gray, (New York: Macmillan, 1927).

28. S. Haack, *Philosophy of logics*, (Cambridge: Cambridge Univ. Press, 1979).

29. A. Hamfelt, *The Multilevel Structure of Legal Knowledge and its Representation*, Uppsala Theses in Computing Science 8/90, (Uppsala: Uppsala University, 1990).

30. A. Hamfelt, J. Barklund, "An Intelligent Interface to Legal Data Bases—Combining Logic Programming and Hypertext", *proc. Database and Expert Systems Applications*, eds. A. M. Tjoa and R. Wagner, (Vienna: Springer-Verlag, 1990), 56–61.

31. A. Hamfelt, J. Barklund, "Metalevels in Legal Knowledge and their Runnable Representation in Logic", *preproc. Third Intl. Congress—Logica, Informatica, Diritto—Expert Systems in Law* vol. II, ed. A. A. Martino, (1989), 557–76.

32. A. Hamfelt, J. Barklund, "Metaprogramming for Representation of Legal Principles", *proc. Second Workshop on Metaprogramming in Logic*, ed. M. Bruynooghe, (Leuven: Katholieke Universiteit Leuven, 1990), 105–22.

33. A. Hamfelt, Å. Hansson, "A Semiformal Metatheory for Fragmentary and Multilayered Knowledge as an Interactive Metalogic Program", *proc. International Conference on Fifth Generation Computer Systems*, (Tokyo: ICOT, 1992).

34. A. Hamfelt, Å. Hansson, "Metalogic Representation of Stratified Knowledge", *UPMAIL TR 66*, (Uppsala: Comp. Sci. dept., Uppsala University, 1991).

35. A. Hamfelt, Å. Hansson, "Representation of Fragmentary and Multilayered Knowledge—A Semiformal Metatheory as an Interactive Metalogic Program", *UPMAIL TR 68*, (Uppsala: Comp. Sci. dept., Uppsala University, 1991).

36. A. Hamfelt, Å. Hansson, "Representation of Fragmentary and Multilayered Knowledge", *proc. Third Workshop on Metaprogramming in Logic*, (1992).

37. A. Hamfelt, P. Wahlgren, "Datorstödda beslut—Artificiell intelligens och juridik", *IRI-rapport 1988:4*, (Stockholm: Univ. of Stockholm, 1988).

38. W. G. Hammond, "On Analogy and the Ratio Legis", *Readings in Jurisprudence*, ed. J. Hall, (Indianapolis: Bobbs-Merrill, 1938), 568–72.

39. H. L. A. Hart, *The Concept of Law*, (Oxford: Clarendon Press, 1961).

40. I. Hedenius, *Om rätt och moral*, (Stockholm: Tidens förlag, 1941).

41. I. Hedenius, "Values and Duties", *Theoria* 15 (1949): 108–15.

42. P. M. Hill, J. W. Lloyd "Analysis of Meta-Programs", *Meta-Programming in Logic Programming*, eds. H. Abramson and M. H. Rogers, (Cambridge, Mass.: MIT Press, 1989), 23–51.

43. H. Høffding, "Begrebet Analogi", *Kgl. Danske Videnskabernes Selskab. Filosofiske meddelelser* I:4 (1923).

44. J. Horovitz, *Law and Logic*, (Vienna: Springer-Verlag, 1972).
45. A. Kaufmann, "The Ontological Structure of Law", *Natural Law Forum* 8 (1963): 79–96.
46. H. Kelsen, *General Theory of Law and State*, (Cambridge, Mass.: Harvard Univ. Press, 1945).
47. H. Kelsen, "Die Grundlage der Naturrechtslehre", *Österreichische Zeitschrift für öffentliches Recht* 13 (1963): 1–37.
48. H. Kelsen, *Pure Theory of Law* (Los Angeles: Univ. of California Press, 1967).
49. S. C. Kleene, *Introduction to Metamathematics*, (New York: North–Holland, 1980).
50. R. A. Kowalski, "Problems and Promises of Computational Logic", *Computational Logic*, ed. J. W. Lloyd, (Berlin: Springer-Verlag, 1990), 1–36.
51. R. A. Kowalski, "The Treatment of Negation in Logic Programs for Representing Legislation", *proc. Second Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1989), 11–5.
52. R. A. Kowalski, M. J. Sergot, "The Use of Logical Models in Legal Problem Solving", *Ratio Juris* vol. 3, no. 2, (1990): 201–18.
53. S. Kripke, "Outline of a Theory of Truth", *J. of Philosophy* (1975): 690–716.
54. L. T. McCarty, "A Language for Legal Discourse I. Basic Features", *proc. Second Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1989), 180–9.
55. L. T. McCarty, "On the Role of Prototypes in Appellate Legal Argument", *proc. Third Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1991), 185–90.
56. K. Nitta, J. Nagao, T. Mizutori, "A Knowledge Representation and Inference System for Procedural Law", *J. New Generation Computing* 5 (1988): 319–59.
57. H. Oliphant, "A Return to Stare Decisis", *Readings in Jurisprudence*, ed. J. Hall, (Indianapolis: Bobbs-Merrill, 1938), 580–3.
58. G. G. Priest, "Modality as a Meta-Concept", *Notre Dame J. of Formal Logic* 17 (1976).
59. A. Peczenik, *Rättsnormer*, (Stockholm: Norstedts, 1987).
60. J. Raz, *The Concept of a Legal System*, (Oxford: Clarendon Press, 1970).
61. A. Ross, "Imperatives and Logic", *Theoria* 7 (1941): 53–71.
62. A. Ross, *Directives and Norms*, (London: Routledge and Kegan Paul, 1969).
63. T. Routen, "Hierarchically Organised Formalisations", *proc. Second Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1989), 242–50.
64. B. Russell, *Introduction to Mathematical Philosophy*, (London: G. Allen and Unwin, 1919).

65. M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, H. T. Cory, "The British Nationality Act as a Logic Program", *Comm. ACM* 29 (May 1986): 370–86.

66. M. J. Sergot, "A Query-the-User Facility for Logic Programming", *Integrated Interactive Computer Systems*, eds. P. Degano and E. Sandewall, (Amsterdam: North-Holland, 1983), 27–41.

67. M. J. Sergot, "The Representation of Law in Computer Programs", *Knowledge-Based Systems and Legal Applications*, ed. T. Bench-Capon, (London: Academic Press, 1991), 3–67.

68. L. Sterling, "Logical Levels of Problem Solving", *J. Logic Programming* vol. 1, no. 2 (1984): 151–63.

69. J. Sticklen, B. Chandrasekaran, and J.R. Josephson, "Control Issues in Classificatory Diagnosis", *proc. Ninth Intl. Joint Conf. on Artificial Intelligence*, (Los Altos, Calif.: Morgan Kaufmann, 1985), 300–6.

70. S. Strömholm, *Rätt, rättskällor och rättstillämpning*, (Stockholm: Norstedts, 1984).

71. J. Sundberg, *fr Eddan t Ekelöf*, (Lund: Studentlitt., 1978).

72. N. K. Sundby, *Om normer*, Filosofiske Problemer XLIII, ed. A. Næss, Scandinavian Studies in Philosophy, (Oslo-Bergen-Tromsø: Universitetsforlaget, 1974).

73. A. Tarski, "The Semantic Conception of Truth", *Philosophy and Phenomenological Research* 4 (1944): 341–75.

74. A. Tarski, *Logic, Semantics and Metamathematics*, trans. J. H. Woodger, (Oxford: Clarendon Press, 1956).

75. A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem", *proc. Lond. Math. Soc., Ser. 2*, 42 (1936), 230–65; correction, *ibid.*, 43 (1937), 544–6.

76. A. Wedberg, *Filosofins historia. Antiken och medeltiden.*, second revised ed., (Stockholm: Bonniers, 1968).

77. R. W. Weyhrauch, "Prolegomena to a Theory of Mechanized Formal Reasoning", *Artificial Intelligence* 13 (1980): 133–70.

78. D. E. Wolstenholme, "Amalgamating Regulation- and Case-Based Advice Systems through Suggested Answers", *proc. Second Intl. Conf. on Artificial Intelligence and Law*, (New York: ACM, 1989), 63–7.

79. G. H. von Wright, "Deontic Logic", *Mind* 60 (1951): 1–15.

80. G. H. von Wright, *Logik, filosofi och språk*, (Stockholm: Aldus/Bonniers, 1965).

81. G. H. von Wright, "Deontic Logic and the Theory of Conditions", *Deontic Logic: Introductory and Systematic Readings*, ed. R. Hilpinen, (Dordrecht: D. Reidel Publ. Co., 1971), 159–77.

UPPSALA THESES IN COMPUTING SCIENCE
15

Metalogic Representation of Multilayered Knowledge

by

Andreas Hamfelt

UPMAIL
Computing Science Department
Uppsala University

UPPSALA 1992

Doctoral thesis at Uppsala University 1992

Abstract

Hamfelt, A., 1992. Metalogic Representation of Multilayered Knowledge. *Uppsala Theses in Computing Science* 15. 125 pp. Uppsala. ISBN 91–506–0917–3. ISSN 0283–359X.

Metalogic programming is an important technique for the three interrelated topics 'knowledge representation', 'knowledge processing', and 'knowledge assimilation'. Its expressiveness enables a logically pure structure preserving representation of multilayered, fragmentary knowledge which, since neither fully formalizable nor static, requires the assimilation of externally supplied knowledge and the ability to cope with change. Typically, formalization of exact theories involves three distinct theories: the informal theory to be formalized, the formal theory, and the informal metatheory discussing the latter of these theories. For multilayered and imprecise theories, we propose instead a semiformal metatheory discussing both theories. Our semiformal metatheory is implemented as a Horn clause metaprogram that interactively constructs and assesses first-order formal theories as proposed representations of a hierarchical informal theory. Presentation of metaproofs (proof terms) allows the user not only to assess the metaprogram's proposals for sentences for the formal theory, but also its reasoning leading to these proposals. Changes in the dynamic domain knowledge are smoothly coped with, since the metaprogram gives a modular representation which is structure preserving both horizontally and vertically. Our sample informal theory is a description of legal knowledge as a hierarchy in which adjacent levels communicate by informal upward reflection. Rules proposed for inclusion in a theory of legally acceptable rules at a level i must be accepted by (meta)rules of legal interpretation in a theory of legally acceptable metarules at level $i + 1$ in turn subject to metametarules at level $i + 2$, etc. As a further complication only schematic descriptions of these rules are available from which case-specific rules must be dynamically proposed for each particular adjudication. Reasoning with imprecise knowledge allows multiple interpretations both semantically and structurally. Our metaprogram is a suggestion for how to account for this: it formalizes multiple semantic interpretations of statute law.

The research reported herein has been supported by the Swedish Board for Technical Development (STU).

Andreas Hamfelt, UPMAIL, Computing Science Department, Uppsala University, Box 520, S-751 20 Uppsala, Sweden.

Typeset in TEX by Jonas Barklund. Printed in Sweden 1992 by Reprocentralen HSC, Uppsala.

For my parents

ACKNOWLEDGEMENTS

This dissertation has been prepared under the supervision of Professor Åke Hansson to whom I am deeply grateful for his constant inspiration and support. The time and effort he has spent on providing careful guidance and constructive suggestions simply cannot be overestimated. Moreover, my work over the years with Dr. Jonas Barklund has been very rewarding. The research presented herein has its origin in a fruitful and inspiring collaboration we initiated around Christmas 1988 and which has continued along several lines ever since.

Professor Sten-Åke Tärnlund has been a constant source of advice, encouragement and fruitful contacts. One example of a rewarding joint activity is our four year engagement in a conference series on legal data processing arranged by the Council of Europe. For access to this forum of knowledge exchange, I am also indebted to Börje Alpsten, Head of Planning and Budget Secretariat, Ministry of Justice.

Notwithstanding all his other undertakings, Professor Keith Clark has engaged himself in my research and his continuous and very competent advice has improved the work in several respects. Moreover, I wish to thank Dr. Leon Sterling for a most interesting afternoon spent with him in Kista (Stockholm) last spring, discussing multilayered knowledge representation in logic.

I am indebted to Professor Jon Bing for numerous helpful and pertinent comments he gave as external examiner of my licentiate thesis which constitutes the basis of Part II of the present work. Moreover, I would like to express my gratitude to Professor Peter Seipel for valuable comments on that earlier work.

Participation in the Compulog Network has afforded access to the most recent research results and valuable exchanges with knowledgeable colleagues. I would like to thank Professor Sten-Åke Tärnlund and the external examiner of my thesis, Professor Robert Kowalski, for providing this forum.

Helpful advice and a supportive working atmosphere have always been available at Upmail. I wish to thank all those, too numerous to name, who have contributed to my work in these respects.

Finally, yet another debt of gratitude is due to Dr. Jonas Barklund who prepared the final typesetting of this manuscript with great talent.

Uppsala, April 24, 1992 — Andreas Hamfelt

CONTENTS

INTRODUCTORY REMARKS

PART I.
SEMIFORMAL METALOGIC MULTILAYERED PROGRAMMING

INDEX