logics

MDPI

Article

Modelling Value-Oriented Legal Reasoning in LogiKEy

Christoph Benzmüller [1,2,*], David Fuenmayor [1,2] and Bertram Lomfeld [3]

[1] AI Systems Engineering, University of Bamberg, 96045 Bamberg, Germany;
david.fuenmayor@uni-bamberg.de or david.fuenmayor@fu-berlin.de
[2] Department of Mathematics and Computer Science, Freie Universität Berlin, 14195 Berlin, Germany
[3] Department of Law, Freie Universität Berlin, 14195 Berlin, Germany; bertram.lomfeld@fu-berlin.de
[*] Correspondence: christoph.benzmueller@uni-bamberg.de or c.benzmueller@fu-berlin.de;
Tel.: +49-(0)951/863-2942

Abstract: The logico-pluralist LogiKEy knowledge engineering methodology and framework is applied to the modelling of a theory of legal balancing, in which legal knowledge (cases and laws) is encoded by utilising context-dependent value preferences. The theory obtained is then used to formalise, automatically evaluate, and reconstruct illustrative property law cases (involving the appropriation of wild animals) within the *Isabelle/HOL* proof assistant system, illustrating how LogiKEy can harness interactive and automated theorem-proving technology to provide a testbed for the development and formal verification of legal domain-specific languages and theories. Modelling value-oriented legal reasoning in that framework, we establish novel bridges between the latest research in knowledge representation and reasoning in non-classical logics, automated theorem proving, and applications in legal reasoning.

Keywords: legal balancing; value-oriented reasoning; automated theorem proving; logical pluralism; proof assistants; Isabelle/HOL

1. Introduction

Law today has to reflect highly pluralistic environments in which a plurality of values, world views and logics coexist. One function of modern, reflexive law is to enable the social interaction within and between such worlds (Teubner [1], Lomfeld [2]). Any sound model of legal reasoning needs to be pluralistic, supporting different value systems, value preferences, and maybe even different logical notions, while at the same time reflecting the uniting force of law.

Adopting such a perspective, in this paper, we apply the logico-pluralistic LogiKEy knowledge engineering methodology and framework (Benzmüller et al. [3]) to the modelling of a theory of value-based legal balancing, a *discoursive grammar* of justification (Lomfeld [4]), which we then employ to formally reconstruct and automatically assess, using the *Isabelle/HOL* proof assistant system, some illustrative property law cases involving the appropriation of wild animals (termed "wild animal cases"; cf. Bench-Capon and Sartor [5], Berman and Hafner [6], and Merrill and Smith [7] (Ch. II. A.1) for background). Lomfeld's *discoursive grammar* is encoded, for our purposes, as a logic-based domain-specific language (DSL), in which the legal knowledge embodied in statutes and case corpora becomes represented as *context-dependent* preferences among (combinations of) values constituting a pluralistic value system or ontology. This knowledge can thus be complemented by further legal and world knowledge, e.g., from legal ontologies (Casanovas et al. [8], Hoekstra et al. [9]).

The LogiKEy framework supports plurality at different layers; cf. Figure 1. Classical higher-order logic (HOL) is fixed as a *universal meta-logic* (Benzmüller [10]) at the base layer (L0), on top of which a plurality of (combinations of) object logics can become encoded (layer L1). Employing these logical notions, we can now articulate a variety of logic-based domain-specific languages (DSLs), theories and ontologies at the next layer (L2),

check for updates

Citation: Benzmüller, C.; Fuenmayor, D.; Lomfeld, B. Modelling Value-Oriented Legal Reasoning in LogiKEy. *Logics* **2024**, 2, 31–78. https://doi.org/10.3390/logics2010003

Academic Editors: Valentin Goranko and Giuseppe Primiero

Received: 13 December 2023
Revised: 5 March 2024
Accepted: 11 March 2024
Published: 14 March 2024

thus enabling the modelling and automated assessment of different application scenarios (layer L3). These linked layers, as featured in the LogiKEy approach, facilitate fruitful interdisciplinary collaboration between specialists in different AI-related domains and domain experts in the design and development of knowledge-based systems.

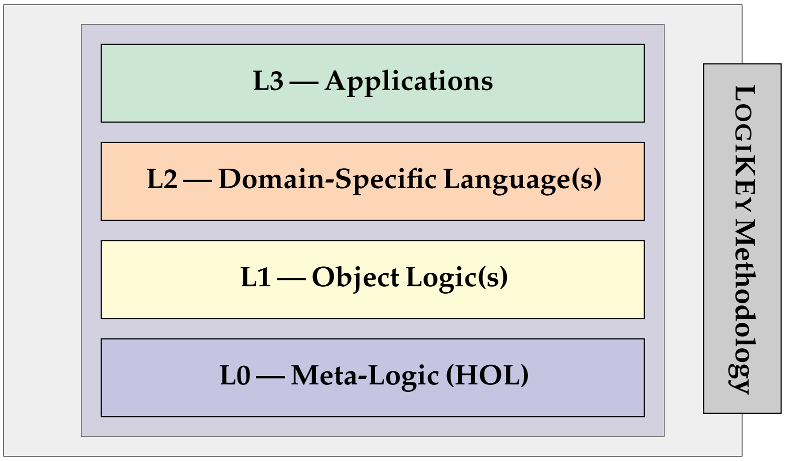


Figure 1. LogiKEy development methodology.

LogiKEy, in this sense, fosters a *division of labour* among different specialist roles. For example, 'logic theorists' can concentrate on investigating the semantics and proof calculi for different object logics, while 'logic engineers' (e.g., with a computer science background) can focus on the encoding of suitable combinations of those formalisms in the meta-logic HOL and on the development and/or integration of relevant automated reasoning technology. Knowledge engineers can then employ these object logics for knowledge representation (by developing ontologies, taxonomies, controlled languages, etc.), while domain experts (ethicists, lawyers, etc.) collaborate with requirements elicitation and analysis, as well as providing domain-specific counselling and feedback. These tasks can be supported in an integrated fashion by harnessing (and extending) modern mathematical proof assistant systems (i.e., interactive theorem provers), which thus become a testbed for the development of logics and ethico-legal theories.

The work reported below is a LogiKEy-supported collaborative research effort involving two computer scientists (Benzmüller and Fuenmayor) together with a lawyer and legal philosopher (Lomfeld), who have joined forces with the goal of studying the computer-encoding and automation of a theory of value-based legal balancing: Lomfeld's *discoursive grammar* (Lomfeld [4]). A formally verifiable legal domain-specific language (DSL) has been developed for the encoding of this theory (at LogiKEy's layer L2). This DSL has been built on top of a suitably chosen object-logical language: a modal logic of preferences (at layer L1), by drawing upon the representation and reasoning infrastructure integrated within the proof assistant *Isabelle/HOL* (layer L0). The resulting system is then employed for the assessment of legal cases in property law (at layer L3), which includes the formal modelling of background legal and world knowledge as required to enable the verification of predicted legal case outcomes and the automatic generation of value-oriented logical justifications (backings) for them.

From a wider perspective, LogiKEy aims at supporting the practical development of computational tools for legal and normative reasoning based on formal methods. One of the main drives for its development has been the introduction of automated reasoning techniques for the design, verification (offline and online), and control of intelligent autonomous systems, as a step towards *explicit ethical agents* (Moor [11], Scheutz [12]). The argument here is that ethico-legal control mechanisms (such as ethical governors; cf. Arkin et al. [13]) of intelligent autonomous systems should be understood and designed as knowledge-based systems, where the required ethical and legal knowledge becomes *explicitly* represented as a logical theory, i.e., as a set of formulas (axioms, definitions, and theorems) encoded in a logic. We have set a special focus on the (re-)use of modern proof assistants based on HOL (*Isabelle/HOL*, *HOL-Light*, *HOL4*, etc.) and integrated automated reasoning tools (*theorem provers* and *model generators*) for the interactive development and verification of ethico-legal theories. To carry out the technical work reported in this paper, we have chosen to work with *Isabelle/HOL*, but the essence of our contributions can easily be applied to other proof assistants and automated reasoning systems for HOL.

Technical results concerning, in particular, our *Isabelle/HOL* encoding have been presented at the *International Conference on Interactive Theorem Proving* (ITP 2021) (Benzmüller and Fuenmayor [14]), and earlier ideas have been discussed at the *Workshop on Models of Legal Reasoning* (MLR 2020). In the present paper, we elaborate on these results and provide a more self-contained exposition by giving further background information on Lomfeld's *discoursive grammar*, on the meta-logic HOL, and on the modal logic of preferences by van Benthem et al. [15].

More fundamentally, this paper presents the full picture as framed by the underlying LogiKEy framework and highlights methodological insights, applications, and perspectives relevant to the *AI and Law* community. One of our main motivations is to help build bridges between recent research in knowledge representation and reasoning in non-classical logics, automated theorem proving, and applications in normative and legal reasoning.

Paper Structure

After summarising the domain legal theory of value-based legal balancing (Lomfeld's *discoursive grammar*) in Section 2, we briefly depict the LogiKEy development and knowledge engineering methodology in Section 3, and our meta-logic HOL in Section 4. We then outline our object logic of choice—a (quantified) modal logic of preferences—in Section 5, where we also present its encoding in the meta-logic HOL and formally verify the preservation of meta-theoretical properties using the *Isabelle/HOL* proof assistant. Subsequently, we model *discoursive grammar* in Section 6 and provide a custom-built DSL, which is again formally assessed using *Isabelle/HOL*. As an illustrative application of our framework, we present in Section 7 the formal reconstruction and assessment of well-known example legal cases in property law ("wild animal cases"), together with some considerations regarding the encoding of required legal and world knowledge. Related and further work is addressed in Section 8, and Section 9 concludes the article.

The *Isabelle/HOL* sources of our formalisation work are available at http://logikey.org (accessed on 12 December 2023) under https://github.com/cbenzmueller/LogiKEy/tree/master/Preference-Logics/EncodingLegalBalancing (accessed on 12 December 2023) (*Preference-Logics/EncodingLegalBalancing*). They are also explained in some detail in Appendix A.

2. A Theory of Legal Values: *Discoursive Grammar* of Justification

The case study with which we illustrate the LogiKEy methodology in the present paper consists in the formal encoding and assessment on the computer of a theory of value-based legal balancing as put forward by Lomfeld [4]. This theory proposes a general set of dialectical (socio-legal) values referred to as a *discoursive grammar* of justification discourses. Lomfeld himself has played the role of the domain expert in our joint research, which from a methodological perspective, can be characterised as being both in part theoretical and in part empirical. Lomfeld's primary role has been to provide background legal domain knowledge and to assess the adequacy of our formalisation results, while informing us of relevant conceptual and legal distinctions that needed to be made. In a sense, this created a the legal theory (*discoursive grammar*) and LogiKEy's methodology have been put to the test. This section presents *discoursive grammar* and discusses some of its merits in comparison to related approaches.

Logical reconstructions quite often separate deductive rule application and inductive case-contextual interpretation as completely distinct ways of legal reasoning (cf. the overview in Prakken and Sartor [16]). Understanding the whole process of legal reasoning as an exchange of opposing action-guiding arguments, i.e., practical argumentation (Alexy [17], Feteris [18]), a strict separation between logically distinct ways of legal reasoning breaks down. Yet, a variety of modes of rule-based (Hage [19], Prakken [20], Modgil and Prakken [21]), case-based (Ashley [22], Aleven [23], Horty [24]) and value-based (Berman and Hafner [6], Bench-Capon et al. [25], Grabmair [26]) reasoning coexist in legal theory and (court) practice.

In line with current computational models combining these different modes of reasoning (e.g., Bench-Capon and Sartor [5], Maranhão and Sartor [27]), we argue that a discourse theory of law can consistently integrate them in the form of a multi-level system of legal reasoning. Legal rules or case precedents can thus be translated into (or analysed as) the balancing of plural and opposing (socio-legal) values on a deeper level of reasoning (Lomfeld [28]).

There exist indeed some models for quantifying legal balancing, i.e., for weighing competing reasons in a case (e.g., Alexy [29], Sartor [30]). We share the opinion that these approaches need to get "integrated with logic and argumentation to provide a comprehensive account of value-oriented reasoning" (Sartor [31]). Hence, a suitable model of legal balancing would need to reconstruct rule subsumption and case distinction as argumentation processes involving conflicting values.

Here, the functional differentiation of legal norms into *rules* and *principles* reveals its potential (Dworkin [32], Alexy [33]). Whereas legal rules have a binary all-or-nothing validity driving out conflicting rules, legal principles allow for a scalable dimension of weight. Thus, principles could outweigh each other without rebutting the normative validity of colliding principles. Legal principles can be understood as a set of plural and conflicting values on a deep level of socio-legal balancing, which is structured by legal rules on an explicit and more concrete level of legal reasoning (Lomfeld [28]). The two-faceted argumentative relation is partly mirrored in the functional differentiation between *goal-norms* and *action-norms* (Sartor [30]) but should not be mixed up with a general understanding of principles as abstract rules (Raz [34], Verheij et al. [35]) or as specific constitutional law elements (Neves [36], Barak [37]).

In any event, if preferences between defeasible rules are reconstructed and justified in terms of preferences between underlying values, some questions about values necessarily pop up. In the words of Bench-Capon and Sartor [5]: "Are values scalar? [. . .] Can values be ordered at all? [. . .] How can sets of values be compared? [. . .] Can values conflict so the promotion of their combination is worse than promoting either separately? Can several less important values together overcome a more important value?".

Hence, an encompassing approach for legal reasoning as practical argumentation needs not only a formal reconstruction of the relation between legal values (or principles) and legal rules but also a substantial framework of values (a basic value system) that allows to systematise value comparison and conflicts as a *discoursive grammar* (Lomfeld [4,28]) of argumentation. In this article, we define a value system not as a "preference order on sets of values" (van der Weide et al. [38]) but as a pluralistic set of values which allow for different preference orders. The computational conceptualisation (as a formal logical theory) of such a set of representational primitives for a pluralist basic value system can then be considered a value "ontology" (Gruber [39,40], Smith [41]), which of course needs to be complemented by further ontologies for relevant background legal and world knowledge (see Casanovas et al. [8], Hoekstra et al. [9]).

Combining the discourse-theoretical idea that legal reasoning is practical argumentation with a two-faceted model of legal norms, legal *rules* could be logically reconstructed as conditional preference relations between conflicting underlying *value principles* (Lomfeld [28], Alexy [33]). The legal consequence of a rule R thus implies the preference of value principle A over value principle B, noted $A > B$ (e.g., health security outweighs freedom to move)[1]. This value preference applies under the condition that the rule's prerequisites E_1 and E_2 hold. Thus, if the propositions E_1 and E_2 are true in a given situation (e.g., a virus pandemic occurs and voluntary shutdown fails), then the value preference

$A > B$ is obtained. This value preference can be said to weight or *balance* the two values A and B against each other. We can thus translate this concrete legal rule as a *conditional preference relation* between colliding value principles:

$$R : (E_1 \land E_2) \Rightarrow A \prec B$$

More generally, A and B could also be structured as aggregates of value principles, whereas the condition of the rule can consist in a conjunction of arbitrary propositions. Moreover, it may well happen that, given some conditions, several rules become relevant in a concrete legal case. In such cases, the rules determine a structure of legal balancing between conflicting plural value principles. Moreover, making explicit the underlying *balancing* of values against each other (as value preferences) helps to justify a legal consequence (e.g., sanctioned lockdown) or ruling in favour of a party (e.g., defendant) in a legal case.

But which value principles are to be balanced? How can we find a suitable justification framework? Based on comparative discourse analyses in different legal systems, one can reconstruct a general dialectical (antagonistic) taxonomy of legal value principles used in (at least Western) legislation, legislative materials, cases, textbooks and scholar writings (Lomfeld [28]). The idea is to provide a plural and yet consistent system of basic legal values and principles (a *discoursive grammar*), independent of concrete cases or legal fields, to justify legal decisions.

The proposed legal value system (Lomfeld [4]), see Figure 2, is consistent with many existing taxonomies of antagonistic psychological (Rokeach [42], Schwartz [43]), political (Eysenck [44], Mitchell [45]), and economic values (Clark [46])[2] . In all social orders, one can observe a general antinomy between individual and collective values. Ideal types of this fundamental normative dialectic are the basic value of FREEDOM for the individual, and the basic value of SECURITY for the collective perspective. Another classic social value antinomy is between a functional–economic (utilitarian) and a more idealistic (egalitarian) viewpoint, represented in the ethical debate by the essential dialectic concerning the basic values of UTILITY versus EQUALITY. These four normative poles stretch two axes of value coordinates for the general value system construction. We thus speak of a normative dialectics since each of the antagonistic basic values and related principles can (and in most situations will) collide with each other.

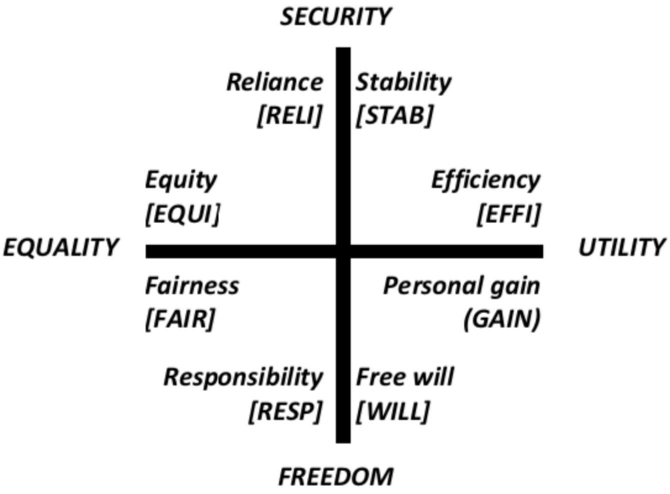


Figure 2. Basic legal value system (ontology) by Lomfeld [4].

Within this dialectical matrix, eight more concrete legal *value principles* are identified. FREEDOM represents the normative basic value of individual autonomy and comprises the legal (value) principles of (more functional) individual choice or 'free will' (WILL) and (more idealistic) (self-)‘responsibility’ (RESP). The basic value of SECURITY addresses the collective dimension of public order and comprises the legal principles of the (more functional) collective ‘stability’ (STAB) of a social system and (more idealistic) social trust or ‘reliance’ (RELI). The value of UTILITY means economic welfare on the personal and collective levels and comprises the legal principles of collective overall welfare maximisation, i.e., ‘efficiency’ (EFFI), and individual welfare maximisation, i.e., economic benefit or ‘gain’ (GAIN). Finally, EQUALITY represents the normative ideal of equal treatment and equal allocation of resources and comprises the legal principles of (more individual) equal opportunity or procedural ‘fairness’ (FAIR) and (more collective) distributional equality or ‘equity’ (EQUI).

The general value system (or ontology) of the proposed *discoursive grammar* can consistently embed existing AI and Law value sets. Earlier accounts of value-oriented reasoning are all tailoring distinct domain value sets for specific case areas. The most prominent and widespread examples are common law property cases concerning the appropriation of (wild) animals (e.g., Berman and Hafner [6], Bench-Capon et al. [25], Sartor [50], Prakken [51]) and its modern variant of a straying baseball (e.g., Bench-Capon [52], Gordon and Walton [53]). In both settings, the defendant’s claim to link property to actual possession is justified with the stability value of (legal) certainty and contrasted with the liberal idea to protect individual activities from (legal) interference favouring the plaintiff (Berman and Hafner [6], Bench-Capon [52]). The *discoursive grammar* reconstructs this normative tension as collision between the general values of SECURITY and FREEDOM. The underlying dialectic reappears in many typical constitutional right cases, where individual liberty collides with collective security issues (Sartor [30]), e.g., also in the form of privacy versus law enforcement in the analysed Fourth Amendment cases (Bench-Capon and Prakken [54]).

The other dialectic dimension of UTILITY v EQUALITY is also represented in the AI and Law value reconstructions. ’UTILITY’ (Bench-Capon [52]) could be understood to embrace the values of ’competition’ (Berman and Hafner [6]), ’economic activity’ (Bench-Capon et al. [25]), ’economic benefit’ (Prakken [51]), and economic ’productivity’ (Sartor [50]). On the other hand, ’EQUALITY’ is addressed with values of ’fairness’, ’equity’, and ’public order’ (Berman and Hafner [6], Bench-Capon [52], Gordon and Walton [53]).

The *discoursive grammar* taxonomy works as well within another classic field of AI and Law reconstruction, i.e., trade secret law (Chorley and Bench-Capon [55]), following the long-standing HYPO modelling tradition (Ashley [22], Bench-Capon [56]). The two-dimensional, four-pole matrix also covers the basic trade secret domain values (Grabmair [26]): ’property’ (SECURITY, UTILITY) and ’confidentiality’ (SECURITY) interests versus free and equal public access to information (FREEDOM, EQUALITY) and open competition (UTILITY, FREEDOM).

A key feature of the dialectical *discoursive grammar* approach to value argumentation consists in its purely qualitative modelling of legal balancing in terms of context-dependent logic-based preferences among values, without any need for (but the possibility of) determining quantitative weights. The modelling is thus more flexible than more hierarchical approaches to value representation (Verheij [57]).

3. The LogiKEy Methodology

LogiKEy, as a methodology (Benzmüller et al. [3]), refers to the principles underlying the organisation and the conduct of complex knowledge design and engineering processes, with a particular focus on the legal and ethical domain. Knowledge engineering refers to all the technical and scientific aspects involved in building, maintaining and using knowledge-based systems, employing logical formalisms as a representation language. In particular, we speak of *logic engineering* to highlight those tasks directly related to the syntactic and semantic definition, as well as to the meta-logical encoding and automation, of different combinations of object logics. It is also LogiKEy’s objective to fruitfully integrate contributions from different research communities (such as interactive and automated theorem proving, non-classical logics, knowledge representation, and domain specialists) and to make them accessible at a suitable level of abstraction and technicality to practitioners in diverse fields.

A fundamental characteristic of the LogiKEy methodology consists in the utilisation of classical higher-order logic (HOL, cf. Benzmüller and Andrews [58]) as a general-purpose logical formalism in which to encode different (combinations of) object logics. This enabling technique is known as shallow[3] semantical embeddings (SSEs). HOL thus acts as the substrate in which a plurality of logical languages, organised hierarchically at different abstraction layers, become ultimately encoded and reasoned with. This in turn enables the provision of powerful tool support: we can harness mathematical proof assistants (e.g., *Isabelle/HOL*) as a testbed for the development of logics, and ethico-legal DSLs and theories. More concretely, off-the-shelf theorem provers and (counter-)model generators for HOL as provided, for example, in the interactive proof assistant *Isabelle/HOL* (Blanchette et al. [61]), are assisting the LogiKEy knowledge and logic engineers (as well as domain experts) to *flexibly experiment* with underlying (object) logics and their combinations, with general and domain knowledge, and with concrete use cases—all at the same time. Thus, continuous improvements of these off-the-shelf provers, without further ado, leverage the reasoning performance in LogiKEy.

The LogiKEy methodology, cf. Figure 1, has been instantiated in this article to support and guide the simultaneous development of the different modelling layers as depicted in Figure 3, which will be the subject of discussion in the following sections. According to the logico-pluralistic nature of LogiKEy, only the lowest layer (L0), meta-logic HOL (cf. Section 4), remains fixed, while all other layers are subject to dynamic adjustments until a satisfying overall solution in the overall modelling process is reached. At the next layer (L1), we are faced with the choice of an object logic, in our case, a modal logic of preference (cf. Section 5). A legal DSL (cf. Section 6), created after *discoursive grammar* (cf. Section 2), further extends this object logic at a higher level of abstraction (layer L2). At the upper layer (layer L3), we use this legal DSL to encode and automatically assess some example legal cases ("wild animal cases") in property law (cf. Section 7) by relying upon previously encoded background legal and world knowledge.[4] The higher layers thus make use of the concepts introduced at the lower layers. Moreover, at each layer, the encoding efforts are guided by selected tests and 'sanity checks' in order to formally verify relevant properties of the encodings at and up to that level.

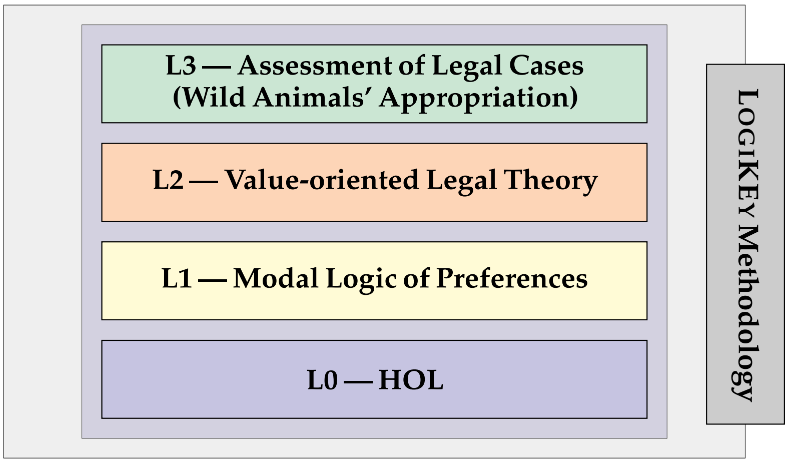

L3 — Assessment of Legal Cases (Wild Animals' Appropriation)
L2 — Value-oriented Legal Theory
L1 — Modal Logic of Preferences
L0 — HOL
LogiKEy Methodology

Figure 3. LogiKEy development methodology as instantiated in the given context.

It is worth noting that the application of our approach to deciding concrete legal cases reflects ideas in the AI and Law literature about understanding the solution of legal cases as theory construction, i.e., "building, evaluating and using theories" (Bench-Capon and

Sartor [5], Maranhão and Sartor [27])[5]. This multi-layered, iterative knowledge engineering process is supported in our LogiKEy framework by adapting interactive and automated reasoning technology for HOL (as a meta-logic).

An important aspect thereby is that the LogiKEy methodology foresees and enables the knowledge engineer to flexibly switch between the modelling layers and to suitably adapt the encodings also at lower layers if needed. The engineering process thus has backtracking points, and several work cycles may be required; thereby, the higher layers may also pose modification requests to the lower layers. Such demands, unlike in most other approaches, may also involve far-reaching modifications of the chosen logical foundations, e.g., in the particularly chosen modal preference logic.

The work we present in this article is, in fact, the result of an iterative, give-and-take process encompassing several cycles of modelling, assessment, and testing activities, whereby a (modular) logical theory gradually evolves until eventually reaching a state of highest coherence and acceptability. In line with previous work on *computational hermeneutics* (Fuenmayor and Benzmüller [63]), we may then speak of arriving at a state of *reflective equilibrium* (Daniels [64]) as the end-point of an iterative process of mutual adjustment among (general) principles and (particular) judgements, where the latter are intended to become logically entailed by the former. A similar idea of *reflective equilibrium* has been introduced by the philosopher John Rawls in moral and political philosophy as a method for the development of a consistent *theory of justice* (Rawls [65]). An even earlier formulation of this approach is often credited to the philosopher Nelson Goodman, who proposed the idea of *reflective equilibrium* as a method for the development of (inference rules for) deductive and inductive logical systems (Goodman [66]). Again, the spirit of LogiKEypoints very much in the same direction.

4. Meta-Logic (L0)—Classical Higher-Order Logic

To keep this article sufficiently self-contained, we briefly introduce a classical higher-order logic, termed HOL; more detailed information on HOL and its automation can be found in the literature (Benzmüller and Andrews [58], Andrews [67,68], Benzmüller et al. [69], Benzmüller and Miller [70]).

The notion of HOL used in this article refers to a simply typed logic of functions that has been put forward by Church [71]. Hence, all terms of HOL are assigned a fixed and unique type, commonly written as a subscript (i.e., the term t_α has α as its type). HOL provides λ notation as an elegant and useful means to denote unnamed functions, predicates, and sets; λ notation also supports compositionality, a feature we heavily exploit to obtain elegant, non-recursive encoding definitions for our logic embeddings in the remainder. Types in HOL eliminate paradoxes and inconsistencies.

HOL comes with a set T of *simple types*, which is freely generated from a set of *basic types* $BT \supseteq \{o, \iota\}$ using the function type constructor $\rightarrow$ (written as a right-associative infix operator). For instance, o, $\iota \rightarrow o$ and $\iota \rightarrow \iota \rightarrow \iota$ are types. The type o denotes a two-element set of truth values, and ι denotes a non-empty set of individuals[6]. Further base types may be added as the need arises.

The *terms* of HOL are inductively defined starting from typed constant symbols (C_α) and typed variable symbols (x_α) using λ*-abstraction* ($(\lambda x_\alpha.\, s_\beta)_{\alpha\rightarrow\beta}$) and *function application* ($(s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta$), thereby obeying type constraints as indicated. Type subscripts and parentheses are usually omitted to improve readability, if obvious from the context or irrelevant. Observe that λ abstractions introduce unnamed functions. For example, the function that adds two to a given argument x can be encoded as $(\lambda x.\, x + 2)$, and the function that adds two numbers can be encoded as $(\lambda x.\, (\lambda y.\, x + y))$[7]. HOL terms of type o are also called formulas[8].

Moreover, to obtain a proper logic, we add $\neg_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o}$ and $\Pi_{(\alpha\rightarrow o)\rightarrow o}$ (for each type α) as predefined typed constant symbols to our language and call them *primitive logical connectives*. *Binder notation* for quantifiers $\forall x_\alpha\ s_o$ is used as an abbreviation for $\Pi_{(\alpha\rightarrow o)\rightarrow o}\lambda x_\alpha.s_o$.

The *primitive logical connectives* are given a fixed interpretation as usual, and from them, other logical connectives can be introduced as abbreviations. Material implication $s_o \rightarrow t_o$ and existential quantification $\exists x_\alpha s_o$, for example, may be introduced as shortcuts for $\neg s_o \vee t_o$ and $\neg \forall x_\alpha \neg s_o$, respectively. Additionally, *description or choice operators* or *primitive equality* $=_{\alpha \rightarrow \alpha \rightarrow o}$ (for each type α), abbreviated as $=^\alpha$, may be added. Equality can also be defined by exploiting Leibniz' principle, expressing that two objects are equal if they share the same properties.

It is well known that, as a consequence of Gödel's Incompleteness Theorems, HOL with standard semantics is necessarily incomplete. In contrast, theorem-proving in HOL is usually considered with respect to so-called general semantics (or Henkin semantics), in which a meaningful notion of completeness can be achieved (Andrews [68], Henkin [73]). Note that standard models are subsumed by Henkin general models such that valid HOL formulas with respect to general semantics are also valid in the standard sense.

For the purposes of the present article, we shall omit the formal presentation of HOL semantics and of its proof system(s). We fix instead some useful notation for use in the remainder. We write $\mathcal{H} \models^{\mathrm{HOL}} \varphi$ if formula φ of HOL is *true* in a Henkin general model $\mathcal{H}$; $\models^{\mathrm{HOL}} \varphi$ denotes that φ is (Henkin) *valid*, i.e., that $\mathcal{H} \models^{\mathrm{HOL}} \varphi$ for all Henkin models $\mathcal{H}$.

5. Object Logic (L1)—A Modal Logic of Preferences

Adopting the LOGIKEY methodology of Section 3 to support the given formalisation challenge, the first question to be addressed is how to (initially) select the object logic at layer L1. The chosen logic not only must be expressive enough to allow the encoding of knowledge about the law (and the world) as required for the application domain (cf. our case study in Section 7) but must also provide the means to represent the kind of conditional value preferences featured in *discoursive grammar* (as described in Section 2). Importantly, it must also enable the adequate modelling of the notions of value aggregation and conflict as featured in our legal DSL (discussed in Section 6).

Our initial choice has been the family of modal logics of preference presented by van Benthem et al. [15], which we abbreviate by $\mathcal{PL}$ in the remainder. $\mathcal{PL}$ has been put forward as a modal logic framework for the formalisation of preferences which also allows for the modelling of ceteris paribus clauses in the sense of "all other things being equal". This reading goes back to the seminal work of von Wright in the early 1960s (von Wright [74])[9].

$\mathcal{PL}$ appears well suited for effective automation using the SSEs approach, which has been an important selection criterion. This judgment is based on good prior experience with SSEs of related (monadic) modal logic frameworks (Benzmüller and Paulson [75,76]), whose semantics employs accessibility relations between possible worlds/states, just as $\mathcal{PL}$ does. We note, however, that this choice of (a family of) object logics ($\mathcal{PL}$) is just one out of a variety of logical systems which can be encoded as fragments of HOL employing the *shallow semantical embedding* approach; cf. Benzmüller [10]. This approach also allows us 'upgrade' our object logic whenever necessary. In fact, we add quantifiers and conditionals to $\mathcal{PL}$ in Section 5.4. Moreover, we may consider combining $\mathcal{PL}$ with other logics, e.g., with normal modal logics by the mechanisms of *fusion* and *product* (Carnielli and Coniglio [77]), or, more generally, by *algebraic fibring* (Carnielli et al. [78] (Ch. 2–3)). This illustrates a central objective of the LOGIKEY approach, namely that the precise choice of a formalisation logic (i.e., the *object logic* at L1) is to be seen as a parameter.

In the subsections below, we start by informally outlining the family of modal logics of preferences $\mathcal{PL}$ (their formal definition can be found in Appendix A.1). We then discuss its embedding as a fragment of HOL using the SSE approach. As for Section 4, the technically and formally less interested reader may actually skip the content of these subsections and return later.

5.1. *The Modal Logic of Preferences $\mathcal{PL}$*

We sketch the syntax and semantics of $\mathcal{PL}$ adapting the description from van Benthem et al. [15] (see Appendix A.1 for more details).

The formulas of $\mathcal{PL}$ are inductively defined as follows (where p ranges over a set `Prop` of propositional constant symbols):

$$\varphi, \psi ::= \ \mathtt{p} \mid \varphi \wedge \psi \mid \neg\varphi \mid \Diamond^{\preceq}\varphi \mid \Diamond^{\prec}\varphi \mid \mathbf{E}\varphi$$

As usual in modal logic, van Benthem et al. [15] give $\mathcal{PL}$ a Kripke-style semantics, which models propositions as sets of states or 'worlds'. $\mathcal{PL}$ semantics employs a reflexive and transitive accessibility relation $\preceq$ (respectively, its strict counterpart $\prec$) to define the modal operators in the usual way. This relation is called a *betterness ordering* (between states or 'worlds').

For the sake of self-containedness, we summarize below the semantics of $\mathcal{PL}$.

A preference model $\mathcal{M}$ is a triple $\mathcal{M} = \langle \mathcal{W}, \preceq, \delta \rangle$ where (i) $\mathcal{W}$ is a set of worlds/states; (ii) $\preceq$ is a *betterness relation* (reflexive and transitive) on $\mathcal{W}$, where its strict subrelation $\prec$ is defined as $w \prec v := w \preceq v \wedge v \not\preceq w$ for all $v, w \in \mathcal{W}$ (totality of $\preceq$, i.e., $v \preceq w \vee w \preceq v$, is generally not assumed); and (iii) δ is a standard modal valuation. Below, we show the truth conditions for $\mathcal{PL}$'s modal connectives (the rest being standard):

$$\begin{aligned}
&\mathcal{M}, w \vDash \Diamond^{\preceq}\varphi \text{ iff } \exists v \in \mathcal{W} \text{ such that } w \preceq v \text{ and } \mathcal{M}, v \vDash \varphi\\
&\mathcal{M}, w \vDash \Diamond^{\prec}\varphi \text{ iff } \exists v \in \mathcal{W} \text{ such that } w \prec v \text{ and } \mathcal{M}, v \vDash \varphi\\
&\mathcal{M}, w \vDash \mathbf{E}\varphi \quad \text{iff } \exists v \in \mathcal{W} \text{ such that } \mathcal{M}, v \vDash \varphi
\end{aligned}$$

A formula φ is *true at* world $w \in W$ in model $\mathcal{M}$ if $\mathcal{M}, w \vDash \varphi$. φ is *globally true in* $\mathcal{M}$, denoted $\mathcal{M} \vDash \varphi$, if φ is *true at* every $w \in W$. Moreover, φ is *valid* (in a class of models $\mathcal{K}$) if *globally true in* every $\mathcal{M}$ ($\in \mathcal{K}$), denoted $\vDash_{\mathcal{PL}} \varphi$ ($\vDash_{\mathcal{K}} \varphi$).

Thus, $\Diamond^{\preceq}\varphi$ (respectively, $\Diamond^{\prec}\varphi$) can informally be read as "φ is true in a state that is considered to be at least as good as (respectively, strictly better than) the current state", and $\mathbf{E}\varphi$ can be read as "there is a state where φ is true".

Further, standard connectives such as $\vee$, $\rightarrow$, and $\leftrightarrow$ can also be defined in the usual way. The dual operators $\Box^{\preceq}\varphi$ (respectively, $\Box^{\prec}\varphi$) and $\mathbf{A}\varphi$ can also be defined as $\neg\Diamond^{\preceq}\neg\varphi$ (respectively, $\neg\Diamond^{\prec}\neg\varphi$) and $\neg\mathbf{E}\neg\varphi$.

Readers acquainted with Kripke semantics for modal logic will notice that $\mathcal{PL}$ features normal *S4* and *K4* diamonds operators $\Diamond^{\preceq}$ and $\Diamond^{\prec}$, together with a global existential modality $\mathbf{E}$. We can thus give the usual reading to $\Box$ and $\Diamond$ as *necessity* and *possibility*, respectively.

Moreover, note that since the *strict* betterness relation $\prec$ is not reflexive, it does not hold in general that $\Box^{\prec}\varphi \rightarrow \varphi$ (modal axiom *T*). Hence, we can also give a 'deontic reading' to $\Diamond^{\prec}\varphi$ and $\Box^{\prec}\varphi$; the former could then read as "φ is admissible/permissible" and the latter as "φ is recommended/obligatory". This deontic interpretation can be further strengthened so that the latter entails the former by extending $\mathcal{PL}$ with the postulate $\Box^{\prec}\varphi \rightarrow \Diamond^{\prec}\varphi$ (modal axiom *D*), or alternatively, by postulating the corresponding (meta-logical) semantic condition, namely, that for each state, there exists a strictly better one (*seriality* for $\prec$).

Observe that we use boldface fonts to distinguish standard logical connectives of $\mathcal{PL}$ from their counterparts in HOL.

Similarly, eight different binary connectives for modelling preference statements between propositions can be defined in $\mathcal{PL}$:

$$\preceq_{EE}/\prec_{EE}\ ,\ \preceq_{EA}/\prec_{EA}\ ,\ \preceq_{AE}/\prec_{AE}\ ,\ \preceq_{AA}/\prec_{AA}\,.$$

These connectives arise from four different ways of 'lifting' the *betterness ordering* $\preceq$ (respectively, $\prec$) on states to a *preference ordering* on sets of states or propositions:

$$\begin{aligned}
(\varphi \preceq_{EE}/\prec_{EE} \psi)\ u \ &\text{ iff } \ \exists t\ \varphi\, s \wedge (\exists t\ \psi\, t \wedge s \preceq/\prec t)\\
(\varphi \preceq_{EA}/\prec_{EA} \psi)\ u \ &\text{ iff } \ \exists t\ \psi\, t \wedge (\forall s\ \varphi\, s \rightarrow s \preceq/\prec t)\\
(\varphi \preceq_{AE}/\prec_{AE} \psi)\ u \ &\text{ iff } \ \forall s\ \varphi\, s \rightarrow (\exists t\ \psi\, t \wedge s \preceq/\prec t)\\
(\varphi \preceq_{AA}/\prec_{AA} \psi)\ u \ &\text{ iff } \ \forall s\ \varphi\, s \rightarrow (\forall t\ \psi\, t \rightarrow s \preceq/\prec t)
\end{aligned}$$

Thus, different choices for a *logic of preference* are possible if we restrict ourselves to employing only a selected preference connective, where each choice provides the logic with particular characteristics so that we can interpret preference statements between propositions (i.e., sets of states) in a variety of ways. As an illustration, according to the semantic interpretation provided by van Benthem et al. [15], we can read $\varphi{\prec_{AA}}\psi$ as "every ψ-state being better than every φ-state", and read $\varphi{\prec_{AE}}\psi$ as "every φ-state having a better ψ-state" (and analogously for others).

In fact, the family of preference logics $\mathcal{PL}$ can be seen as encompassing, in substance, the proposals by von Wright [74] (variant $\prec_{AA}$) and Halpern [79] (variants $\preceq_{AE}/\prec_{AE}$)[10]. As we will see later in Section 6, there are only four choices ($\preceq_{EA}/\prec_{EA}$ and $\preceq_{AE}/\prec_{AE}$) of modal preference relations that satisfy the minimal conditions we impose for a logic of value aggregation. Moreover, they are the only ones which satisfy transitivity, a quite controversial property in the literature on preferences.

Last but not least, van Benthem et al. [15] have provided 'syntactic' counterparts for these binary preference connectives as derived operators in the language of $\mathcal{PL}$ (i.e., defined by employing the modal operators $\Diamond^{\preceq}\varphi$ (respectively, $\Diamond^{\prec}\varphi$). We note these 'syntactic variants' in boldface font:

$$\begin{array}{lcl}
(\varphi \preceq_{EE} \psi) := \boldsymbol{E}(\varphi \wedge \Diamond^{\preceq}\psi) & \text{and} & (\varphi \prec_{EE} \psi) := \boldsymbol{E}(\varphi \wedge \Diamond^{\prec}\psi) \\
(\varphi \preceq_{EA} \psi) := \boldsymbol{E}(\psi \wedge \Box^{\prec}\neg\varphi) & \text{and} & (\varphi \prec_{EA} \psi) := \boldsymbol{E}(\psi \wedge \Box^{\preceq}\neg\varphi) \\
(\varphi \preceq_{AE} \psi) := \boldsymbol{A}(\varphi \rightarrow \Diamond^{\preceq}\psi) & \text{and} & (\varphi \prec_{AE} \psi) := \boldsymbol{A}(\varphi \rightarrow \Diamond^{\prec}\psi) \\
(\varphi \preceq_{AA} \psi) := \boldsymbol{A}(\psi \rightarrow \Box^{\prec}\neg\varphi) & \text{and} & (\varphi \prec_{AA} \psi) := \boldsymbol{A}(\psi \rightarrow \Box^{\preceq}\neg\varphi_{\sigma})
\end{array}$$

The relationship between both, i.e., the semantically and syntactically defined families of binary preference connectives is discussed in van Benthem et al. [15]. In a nutshell, as regards the *EE* and the *AE* variants, both definitions (syntactic and semantic) are equivalent; concerning the *EA* and the *AA* variants, the equivalence only holds for a total $\preceq$ relation. In fact, drawing upon our encoding of $\mathcal{PL}$ as presented in the next subsection Section 5.2, we have employed *Isabelle/HOL* for automatically verifying this sort of meta-theoretic correspondence; cf. Lines 4–12 in Figure A4 in Appendix A.1.

5.2. Embedding $\mathcal{PL}$ in HOL

For the implementation of $\mathcal{PL}$ we utilise the *shallow semantical embeddings* (SSE) technique, which encodes the language constituents of an object logic, $\mathcal{PL}$ in our case, as expressions (λ-terms) in HOL. A core idea is to model (relevant parts of) the semantical structures of the object logic explicitly in HOL. This essentially shows that the object logic can be unravelled as a fragment of HOL and hence automated as such. For (multi-)modal normal logics, like $\mathcal{PL}$, the relevant semantical structures are relational frames constituted by sets of possible worlds/states and their accessibility relations. $\mathcal{PL}$ formulas can thus be encoded as predicates in HOL taking possible worlds/states as arguments[11]. The detailed SSE of the basic operators of $\mathcal{PL}$ in HOL is presented and formally tested in Appendix A.1. Further extensions to support reasoning with ceteris paribus clauses in $\mathcal{PL}$ are studied there as well.

As a result, we obtain a combined interactive and automated theorem prover and model finder for $\mathcal{PL}$ (and its extensions; cf. Section 5.4) realised within *Isabelle/HOL*. This is a new contribution since we are not aware of any other existing implementation and automation of such a logic. Moreover, as we will demonstrate below, the SSE technique supports the automated assessment of the meta-logical properties of the embedded logic in *Isabelle/HOL*, which in turn provides practical evidence for the correctness of our encoding.

The embedding starts out with declaring the HOL base type ι, which is denoting the set of possible states (or worlds) in our formalisation. $\mathcal{PL}$ propositions are modelled as predicates on objects of type ι (i.e., as *truth sets* of states/worlds) and, hence, they are given the type $\iota \rightarrow o$, which is abbreviated as σ in the remainder. The *betterness relation* $\preceq$ of $\mathcal{PL}$ is introduced as an uninterpreted constant symbol $\preceq_{\iota\rightarrow\iota\rightarrow o}$ in HOL, and its strict variant $\prec$ is introduced as an abbreviation $\prec_{\iota\to\iota\to o}$ standing for the HOL term $\lambda v.\,\lambda w.\,(v \leq w \wedge \neg(w \leq v))$. In accordance with van Benthem et al. [15], we postulate that $\preceq$ is a preorder, i.e., reflexive and transitive.

In a next step, the σ-type lifted logical connectives of $\mathcal{PL}$ are introduced as abbreviations for λ-terms in the meta-logic HOL. The conjunction operator $\wedge$ of $\mathcal{PL}$, for example, is introduced as an abbreviation $\wedge_{\sigma\to\sigma\to\sigma}$, which stands for the HOL term $\lambda\varphi_\sigma.\lambda\psi_\sigma.\lambda w_\iota.\,(\varphi\, w\ \wedge\ \psi\, w)$, so that $\varphi_\sigma \wedge \psi_\sigma$ reduces to $\lambda w_\iota.\,(\varphi\, w\ \wedge\ \psi\, w)$, denoting the set[12] of all possible states w in which both φ and ψ hold. Analogously, for the negation, we introduce an abbreviation $\neg_{\sigma\to\sigma}$ which stands for $\lambda\varphi_\sigma.\lambda w_\iota.\,\neg(\varphi\, w)$.

The operators $\Diamond^{\preceq}$ and $\Diamond^{\prec}$ use $\preceq$ and $\prec$ as guards in their definitions. These operators are introduced as shorthand $\Diamond^{\preceq}_{\sigma\to\sigma}$ and $\Diamond^{\prec}_{\sigma\to\sigma}$ abbreviating the HOL terms $\lambda\varphi_\sigma.\lambda w_\iota.\,\exists v_\iota(w \preceq v \wedge \varphi\, v)$ and $\lambda\varphi_\sigma.\lambda w_\iota.\,\exists v_\iota(w \prec v \wedge \varphi\, v)$, respectively. $\Diamond^{\preceq}_{\sigma\to\sigma}\varphi_\sigma$ thus reduces to $\lambda w_\iota.\,\exists v_\iota(w \preceq v \wedge \varphi\, v)$, denoting the set of all worlds w so that φ holds in some world v that is at least as good as w; analogously, for $\Diamond^{\prec}_{\sigma\to\sigma}$. Additionally, the *global existential* modality $\mathbf{E}_{\sigma\to\sigma}$ is introduced as a shorthand for the HOL term $\lambda\varphi_\sigma.\lambda w_\iota.\,\exists v_\iota(\varphi\, v)$. The duals $\Box^{\preceq}_{\sigma\to\sigma}\varphi_\sigma$, $\Box^{\prec}_{\sigma\to\sigma}\varphi_\sigma$ and $\mathbf{A}_{\sigma\to\sigma}\varphi$ can easily be added so that they are equivalent to $\neg\Diamond^{\preceq}_{\sigma\to\sigma}\neg\varphi_\sigma$, $\neg\Diamond^{\prec}_{\sigma\to\sigma}\neg\varphi_\sigma$ and $\neg\mathbf{E}_{\sigma\to\sigma}\neg\varphi$, respectively.

Moreover, a special predicate $\lfloor\varphi_\sigma\rfloor$ (read φ_σ is *valid*) for σ-type lifted $\mathcal{PL}$ formulas in HOL is defined as an abbreviation for the HOL term $\forall w_\iota(\varphi_\sigma\, w)$.

The encoding of object logic $\mathcal{PL}$ in meta-logic HOL is presented in full detail in Appendix A.1.

Remember again that in the LogiKEy methodology, the modeller is not enforced to make an irreversible selection of an object logic (L1) before proceeding with the formalisation work at higher LogiKEy layers. Instead, the framework enables preliminary choices at all layers, which can easily be revised by the modeller later on if this is indicated by, for example, practical experiments.

5.3. Formally Verifying Encoding's Adequacy

A pen-and-paper proof of the faithfulness (soundness and completeness) of the SSE easily follows from previous results regarding the SSE of propositional multi-modal logics (Benzmüller and Paulson [75]) and their quantified extensions (Benzmüller and Paulson [76]); cf. also Benzmüller [10] and the references therein. We sketch such an argument below, as it provides an insight into the underpinnings of SSE for the interested reader.

By drawing upon the approach in Benzmüller and Paulson [75], it is possible to define a mapping between semantic structures of the object logic $\mathcal{PL}$ (preference models $\mathcal{M}$) and the corresponding structures in HOL (general Henkin models $\mathcal{H}^{\mathcal{M}}$) in such a way that

$$\models^{\mathrm{HOL}(\Gamma)} \lfloor\varphi_\sigma\rfloor \quad \text{iff} \quad \models_{\mathcal{PL}} \varphi \quad \text{iff} \quad \vdash_{\mathcal{PL}} \varphi$$

where $\vdash_{\mathcal{PL}}$ denotes derivability in the (complete) calculus axiomatised by van Benthem et al. [15]. Observe that HOL(Γ) corresponds to HOL extended with the relevant types and constants plus a set Γ of axioms encoding $\mathcal{PL}$ semantic conditions, e.g., the reflexivity and transitivity of $\preceq_{\iota\to\iota\to o}$.

The soundness of the SSE (i.e., $\models^{\mathrm{HOL}(\Gamma)} \lfloor\varphi_\sigma\rfloor$ implies that $\models_{\mathcal{PL}} \varphi$) is particularly important since it ensures that our modelling does not give any 'false positives', i.e., proofs of $\mathcal{PL}$ non-theorems. The completeness of the SSE (i.e., $\models_{\mathcal{PL}} \varphi$ implies $\models^{\mathrm{HOL}(\Gamma)} \lfloor\varphi_\sigma\rfloor$) means that our modelling does not give any 'false negatives', i.e., spurious counterexamples. Besides the pen-and-paper proof, completeness can also be mechanically verified by deriving the σ-type lifted $\mathcal{PL}$ axioms and inference rules in HOL(Γ); cf. Figures A4 and A5 in Appendix A.1.

To gain practical evidence of the faithfulness of our SSE of $\mathcal{PL}$ in *Isabelle/HOL* and also to assess the proof automation performance, we have conducted numerous experiments in which we automatically verify the meta-theoretical results on $\mathcal{PL}$ as presented by van

Benthem et al. [15][13]. Note that these statements thus play a role analogous to that of a requirements specification document (cf. Figures A4 and A5 in Appendix A.1).

5.4. Beyond $\mathcal{PL}$: Extending the Encoding with Quantifiers and Conditionals

We can further extend our encoded logic $\mathcal{PL}$ by adding quantifiers. This is performed by identifying $\forall x_\alpha s_\sigma$ with the HOL term $\lambda w_\iota.\forall x_\alpha(s_\sigma w)$ and $\exists x_\alpha s_\sigma$ with $\lambda w_\iota.\exists x_\alpha(s_\sigma w)$; cf. *binder notation* in Section 4. This way, quantified expressions can be seamlessly employed in our modelling at the upper layers (as performed exemplarily in Section 7). We refer the reader to Benzmüller and Paulson [76] for a more detailed discussion (including faithfulness proofs) of SSEs for *quantified* (multi-)modal logics.

Moreover, observe that having a semantics based on *preferential structures* allows us to extend our logic with a (defeasible) conditional connective $\Rightarrow$. This can be performed in several closely related ways. As an illustration, drawing upon an approach by Boutilier [88], we can further extend the SSE of $\mathcal{PL}$ by defining the connective:

$$\varphi_\sigma \Rightarrow \psi_\sigma := \boldsymbol{A}(\varphi_\sigma \rightarrow \Diamond^{\preceq}(\varphi_\sigma \wedge \Box^{\preceq}(\varphi_\sigma \rightarrow \psi_\sigma))).$$

An intuitive reading of this conditional statement would be "every φ-state has a reachable φ-state such that ψ holds there in also in every reachable φ-state" (where we can interpret "reachable" as "at least as good"). This is equivalent, for finite models, to demanding that all 'best' φ states are ψ states; cf. Lewis [89]. This can indeed be shown equivalent to the approach of Halpern [79], who axiomatises a strict binary preference relation $\succ^s$, interpreted as "relative likelihood"[14]. For further discussion regarding the properties and applications of this—and other similar—preference-based conditionals, we refer the interested reader to the discussions in van Benthem [90] and Liu [91] (Ch. 3).

6. Domain Specific Language (L2)—Value-Oriented Legal Theory

In this section, we incrementally define a domain-specific language (DSL) for reasoning with values in a legal context. We start by defining a "logic of value preferences" on top of the object logic $\mathcal{PL}$ (layer L1). This logic is subsequently encoded in *Isabelle/HOL*, and in the process, it becomes suitably extended with custom means to encode the *discoursive grammar* in Section 2. We thus obtain a HOL-based DSL formally modelling *discoursive grammar*. This formally verifiable DSL is then put to the test using theorem provers and model generators.

Recall from the discussion of the *discoursive grammar* in Section 2 that value-oriented legal rules can become expressed as context-dependent preference statements between *value principles* (e.g., *RELIance*, *STABility*, and *WILL*). Moreover, these value principles were informally associated with basic *values* (i.e., *FREEDOM*, *UTILITY*, *SECURITY* and *EQUALITY*), in such a way as to arrange the first over (the quadrants of) a plane generated by two axes labelled by the latter. More specifically, each axis' pole is labelled by a basic value, with values lying at contrary poles playing somehow antagonistic roles (e.g., *FREEDOM* vs. *SECURITY*). We recall the corresponding diagram (Figure 2) below for the sake of illustration.

Inspired by this theory, we model the notion of a (value) *principle* as consisting of a collection (in this case, a set[15]) of *basic values*. Thus, by considering principles as structured entities, we can more easily define adequate notions of aggregation and conflict among them; cf. Section 6.

From a logical point of view, it is additionally required to conceive value principles as truth bearers, i.e., propositions[16]. We thus seem to face a dichotomy between, at the same time, modelling value principles as sets of basic values and modelling them as sets of worlds. In order to adequately tackle this modelling challenge, we make use of the mathematical notion of a *Galois connection*[17].

For the sake of exposition, Galois connections are to be exemplified by the notion of *derivation operators* in the theory of Formal Concept Analysis (FCA), from which we took inspiration; cf. Ganter and Wille [93]. FCA is a mathematical theory of concepts and concept hierarchies as formal ontologies, which finds practical application in many computer science fields, such as data mining, machine learning, knowledge engineering, semantic web, etc.[18].

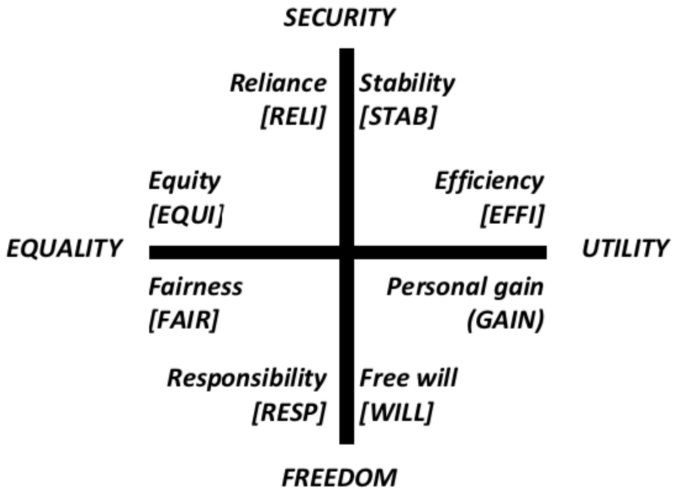


Figure 4. Basic legal value system (ontology) by Lomfeld [4].

6.1. Some Basic FCA Notions

A *formal context* is a triple $K = \langle G, M, I \rangle$, where G is a set of *objects*, M is a set of *attributes*, and I is a relation between G and M (usually called *incidence relation*), i.e., $I \subseteq G \times M$. We read $\langle g, m \rangle \in I$ as "the object g has the attribute m". Additionally, we define two so-called *derivation operators* ↑ and ↓ as follows:

$$A\uparrow \; := \{m \in M \mid \langle g, m \rangle \in I \text{ for all } g \in A\} \qquad \text{for } A \subseteq G \tag{1}$$
$$B\downarrow \; := \{g \in G \mid \langle g, m \rangle \in I \text{ for all } m \in B\} \qquad \text{for } B \subseteq M \tag{2}$$

$A\uparrow$ is the set of all attributes shared by all objects from A, which we call the *intent* of A. Dually, $B\downarrow$ is the set of all objects sharing all attributes from B, which we call the *extent* of B. This pair of derivation operators thus forms an antitone *Galois connection* between (the powersets of) G and M, and we always have that $B \subseteq A\uparrow$ iff $A \subseteq B\downarrow$; cf. Figure 5.

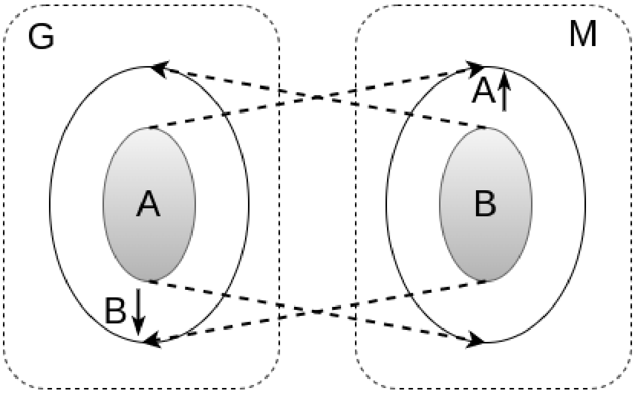


Figure 5. A suggestive representation of a Galois connection between a set of objects G (e.g., worlds) and set of their attributes M (e.g., values).

A *formal concept* (in a context K) is defined as a pair $\langle A, B \rangle$ such that $A \subseteq G$, $B \subseteq M$, $A\uparrow = B$, and $B\downarrow = A$. We call A and B the *extent* and the *intent* of the concept $\langle A, B \rangle$, respectively[19]. Indeed, $\langle A\uparrow\downarrow, A\uparrow \rangle$ and $\langle B\downarrow, B\downarrow\uparrow \rangle$ are always concepts.

The set of concepts in a formal context is partially ordered by set inclusion of their extents, or, dually, by the (reversing) inclusion of their intents. In fact, for a given formal context, this ordering forms a complete lattice: its *concept lattice*. Conversely, it can be shown that every complete lattice is isomorphic to the concept lattice of some formal context. We can thus define lattice-theoretical meet and join operations on FCA concepts in order to obtain an algebra of concepts[20]:

$$\langle A_1, B_1 \rangle \wedge \langle A_2, B_2 \rangle := \langle (A_1 \cap A_2)\,, (B_1 \cup B_2)\downarrow\uparrow \rangle \tag{3}$$

$$\langle A_1, B_1 \rangle \vee \langle A_2, B_2 \rangle := \langle (A_1 \cup A_2)\uparrow\downarrow\,, (B_1 \cap B_2) \rangle \tag{4}$$

6.2. A Logic of Value Preferences

In order to enable the modelling of the legal theory (*discoursive grammar*) as discussed in Section 2, we will enhance our object logic $\mathcal{PL}$ with additional expressive means by drawing upon the FCA notions expounded above and by assuming an *arbitrary* domain set $\mathcal{V}$ of basic values.

A first step towards our legal DSL is to define a pair of operators $\uparrow$ and $\downarrow$ such that they form a Galois connection between the semantic domain $\mathcal{W}$ of worlds/states of $\mathcal{PL}$ (as 'objects' G) and the set of basic values $\mathcal{V}$ (as 'attributes' M). By employing the operators $\uparrow$ and $\downarrow$ in an appropriate way, we can obtain additional well-formed $\mathcal{PL}$ terms, thus converting our object logic $\mathcal{PL}$ in a logic of value preferences[21]. Details follow.

6.2.1. Principles, Values and Propositions

We introduce a *formal context* $K = \langle \mathcal{W}, \mathcal{V}, \mathcal{I} \rangle$ composed by the set of worlds $\mathcal{W}$, the set of basic values $\mathcal{V}$, and the (implicit) relation $\mathcal{I} \subseteq \mathcal{W} \times \mathcal{V}$, which we might interpret, intuitively, in a teleological sense: $\langle w, v \rangle \in \mathcal{I}$ means that value v provides reasons for the situation (world/state) w to obtain.

Now, recall that we aim at modelling value principles as sets of basic values (i.e., elements of $2^{\mathcal{V}}$), while, at the same time, conceiving of them as propositions (elements of $2^{\mathcal{W}}$). Indeed, drawing upon the above FCA notions allows us to overcome this dichotomy. Given the formal context $K = \langle \mathcal{W}, \mathcal{V}, \mathcal{I} \rangle$, we can define the pair of derivation operators $\uparrow$ and $\downarrow$ employing the corresponding definitions ((1)–(2)) above.

We can now employ these derivation operators to switch between the '(value) principles as sets of (basic) values' and the 'principles as propositions (sets of worlds)' perspectives. Hence, we can now—recalling the informal discussion of the semantics of the object logic $\mathcal{PL}$ in Section 5—give an intuitive reading for truth at a world in a preference model to terms of the form $P\downarrow$; namely, we can read $\mathcal{M}, w \models P\downarrow$ as "principle P provides a reason for (state of affairs) w to obtain". In the same vein, we can read $\mathcal{M} \models A \rightarrow P\downarrow$ as "principle P provides a reason for proposition A being the case"[22].

6.2.2. Value Aggregation

Recalling *discoursive grammar*, as discussed in Section 2, our logic of value preferences must provide means for expressing conditional preferences between value principles, according to the schema:

$$E_1 \;\wedge\; \ldots \;\wedge\; E_n \quad \Rightarrow \quad (A_1 \oplus \cdots \oplus A_n) \prec (B_1 \oplus \cdots \oplus B_n)$$

As regards the preference relation (connective $\prec$), we might think that, in principle, any choice among the eight preference relation variants in $\mathcal{PL}$ (cf. Section 5) will work. Let us recall, however, that *discoursive grammar* also presupposed some (no further specified) mechanism for aggregating value principles (operator $\oplus$); thus, the joint selection of both a preference relation and a aggregation operator cannot be arbitrary: they need to interact in an appropriate way. We explore first a suitable mechanism for value aggregation before we get back to this issue.

Suppose that, for example, we are interested in modelling a legal case in which, say, the principle of "respect for property" *together with* the principle "economic benefit for society" *outweighs* the principle of "legal certainty"[23]. A binary connective $\oplus$ for modelling this notion of *together with*, i.e., for aggregating legal principles (as reasons) must, expectedly, satisfy particular logical constraints in interaction with a (suitably selected) value preference relation $\prec$:

$$\begin{aligned} &(A \prec B) \to (A \prec B \oplus C) \text{ but not } (A \prec B \oplus C) \to (A \prec B) && \text{right aggregation} \\ &(A \oplus C \prec B) \to (A \prec B) \text{ but not } (A \prec B) \to (A \oplus C \prec B) && \text{left aggregation} \\ &(B \prec A) \wedge (C \prec A) \to (B \oplus C \prec A) && \text{union property (opt.)} \end{aligned}$$

For our purposes, the aggregation connectives are most conveniently defined using set union (FCA join), which gives us commutativity. As it happens, only the $\prec_{AE}/\preceq_{AE}$ and $\prec_{EA}/\preceq_{EA}$ variants from Section 5 satisfy the first two conditions. They are also the only variants satisfying transitivity. Moreover, if we choose to enforce the optional third aggregation principle (called "union property"; cf. Halpern [79]), then we would be left with only one variant to consider, namely $\prec_{AE}/\preceq_{AE}$[24].

In the end, after extensive computer-supported experiments in *Isabelle/HOL* we identified the following candidate definitions for the value aggregation and preference connectives, which satisfy our modelling desiderata[25] :

- For the binary value aggregation connective $\oplus$, we identified the following two candidates (both taking two value principles and returning a proposition):

  $$A \oplus_{(1)} B := (A \cap B)\downarrow$$
  $$A \oplus_{(2)} B := (A\downarrow \vee B\downarrow)$$

  Observe that $\oplus_1$ is based upon the join operation on the corresponding FCA formal concepts (see Equation (4)). $\oplus_2$ is a strengthening of the first since $(A \oplus_2 B) \subseteq (A \oplus_1 B)$.
- For a binary preference connective $\prec$ between propositions, we have as candidates:

  $$\varphi \prec_{(1)} \psi := \varphi \preceq_{AE} \psi$$
  $$\varphi \prec_{(2)} \psi := \varphi \prec_{AE} \psi$$
  $$\varphi \prec_{(3)} \psi := \varphi \preceq_{EA} \psi$$
  $$\varphi \prec_{(4)} \psi := \varphi \prec_{EA} \psi$$

In line with the LOGIKEY methodology, we consider the concrete choices of definitions for $\prec$, $\oplus$, and even $\Rightarrow$ (classical or defeasible) as parameters in our overall modelling process. No particular determination is enforced in the LOGIKEY approach, and we may alter any preliminary choices as soon as this appears appropriate. In this spirit, we experimented with the listed different definition candidates for our connectives and explored their behaviour. We will present our final selection in Section 6.3.

6.2.3. Promoting Values

Given that we aim at providing a logic of value preferences for use in legal reasoning, we still need to consider the mechanism by which we can link legal decisions, together with other legally relevant facts, to values. We conceive of such a mechanism as a sentence schema, which reads intuitively as "Taking decision D in the presence of facts F *promotes* (value) principle P". The formalisation of this schema can indeed be seen as a new predicate in the domain-specific language (DSL) that we have been gradually defining in this section. In the expression *Promotes(F,D,P)*, we have that F is a conjunction of facts relevant to the case (a proposition), D is the legal decision, and P is the value principle thereby promoted[26]:

$$Promotes(F, D, P) := \quad F \to \Box^{\prec}(D \leftrightarrow \Diamond^{\prec} P\downarrow)$$

It is important to remark that, in the spirit of the LogiKEy methodology, the definition above has arisen from the many iterations of encoding, testing and 'debugging' of the modelling of the 'wild animal cases' in Section 7 (until reaching a *reflective equilibrium*). We can still try to give this definition a somewhat intuitive interpretation, which might read along the lines of "given the facts F, taking decision D is (necessarily) tantamount to (possibly) observing principle P", with the caveat that the (bracketed) modal expressions would need to be read in a non-alethic mood (e.g., deontically as discussed in Section 5.1).

6.2.4. Value Conflict

Another important idea inspired from *discoursive grammar* in Section 2 is the notion of value *conflict*. As discussed there (see Figure 2), values are disposed around two axes of value coordinates, with values lying at contrary poles playing antagonistic roles. For our modelling purposes, it makes thus sense to consider a predicate *Conflict* on worlds (i.e., a proposition) signalling situations where value conflicts appear. Taking inspiration from the traditional logical principle of *ex contradictio sequitur quodlibet*, which we may intuitively paraphrase for the present purposes as *ex conflictio sequitur quodlibet*[27], we define *Conflict* as the set of those worlds in which *all* basic values become applicable:

$$Conflict := \bigwedge\{v\}\downarrow \quad \text{for all } v \text{ in } \mathcal{V}$$

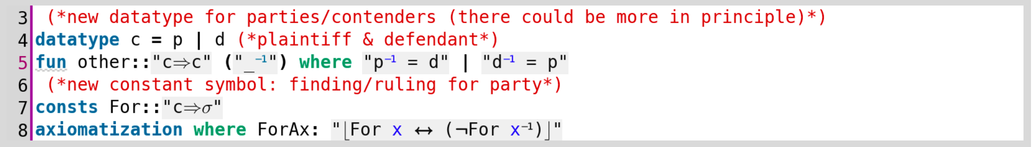
Of course, and in the spirit of the LogiKEy methodology, the specification of such a predicate can be further improved upon by the modeller as the need arises.

6.3. Instantiation as a HOL-Based Legal DSL

In this subsection, we encode our logic of value preferences in HOL (recall discussion in Section 4), building incrementally on top of the corresponding HOL encoding for our (extended) object logic $\mathcal{PL}$ in Section 5.2. In the process, our encoding will be gradually extended with custom means to encode the domain legal theory (cf. *discoursive grammar* in Section 2). For the sake of illustrating a concrete, formally verifiable modelling, we also present in most cases the corresponding encoding in *Isabelle/HOL* (see also Appendix A.2).

In a preliminary step, we introduce a new base HOL-type c (for "contender") as an (extensible) two-valued type introducing the legal parties "plaintiff" (`p`) and "defendant" (`d`). For this, we employ in *Isabelle/HOL* the keyword `datatype`, which has the advantage of automatically generating (under the hood) the adequate axiomatic constraints (i.e., the elements `p` and `d` are distinct and exhaustive).

We also introduce a function, suggestively termed $\texttt{other}_{c\to c}$, with notation $(\cdot)^{-1}$. This function is used to return for a given party the *other* one, i.e., $\texttt{p}^{-1} = \texttt{d}$ and $\texttt{d}^{-1} = \texttt{p}$. Moreover, we add a ($\sigma$-lifted) predicate $\texttt{For}_{c\to\sigma}$ to model the ruling *for* a given party and postulate that it always has to be ruled for either one party or the other: $\texttt{For}\ x \leftrightarrow \neg\texttt{For}\ x^{-1}$.



```
 (*new datatype for parties/contenders (there could be more in principle)*)
datatype c = p | d (*plaintiff & defendant*)
fun other::"c⇒c" ("_⁻¹") where "p⁻¹ = d" | "d⁻¹ = p"
 (*new constant symbol: finding/ruling for party*)
consts For::"c⇒σ"
axiomatization where ForAx: "⌊For x ↔ (¬For x⁻¹)⌋"
```

As a next step, in order to enable the encoding of basic values, we introduce a four-valued datatype $(\text{'}t)\,\texttt{VAL}$ (corresponding to our domain $\mathcal{V}$ of all values). Observe that this datatype is parameterised with a type variable $\text{'}t$. In the remainder, we will always instantiate $\text{'}t$ with the type c (see discussion below):

$$(\text{'}t)\,\texttt{VAL} := \ \texttt{FREEDOM}\,\text{'}t \mid \texttt{UTILITY}\,\text{'}t \mid \texttt{SECURITY}\,\text{'}t \mid \texttt{EQUALITY}\,\text{'}t$$

We also introduce some convenient type aliases.

$v := (c)\,\texttt{VAL} \to o$ is introduced as the type for (characteristic functions of) sets of basic values. The reader will recall that this corresponds to the characterisation of value principles as given in the previous subsection (i.e., elements of $2^{\mathcal{V}}$).

It is important to note, however, that to enable the modelling of legal cases (plaintiff v. defendant), we need to further specify *legal* value principles *with respect to a legal party*, either plaintiff or defendant. For this, we define $cv := c \to v$ intended as the type for specific legal (value) principles (with respect to a legal party) so that they are functions taking objects of type c (either p or d) to sets of basic values.

```
 (*new parameterized datatype for abstract values (wrt. a given party)*)
datatype 't VAL = FREEDOM 't | UTILITY 't | SECURITY 't | EQUALITY 't
type_synonym v = "(c)VAL⇒bool" (*principles: sets of (abstract) values*)
type_synonym cv = "c⇒v" (*principles are specified wrt. a given party*)
```

We introduce useful set-constructor operators for basic values ($\{\ldots\}$) and a superscript notation for specification with respect to a legal party. As an illustration, recalling the discussion in Section 2, we have that, for example, the legal principle of STABility with respect to the plaintiff (notation $\texttt{STAB}^{\texttt{p}}$) can be encoded as a two-element set of basic values (with respect to the plaintiff), i.e., $\{\texttt{SECURITY p}, \texttt{UTILITY p}\}$.

The corresponding *Isabelle/HOL* encoding is as follows.

```
 (*notation for sets*)
abbreviation vset1 ("⦃_⦄") where "⦃φ⦄ ≡ λx::(c)VAL. x=φ"
abbreviation vset2 ("⦃_,_⦄")   where "⦃α,β⦄ ≡ λx::(c)VAL. x=α ∨ x=β"
 (*abstract values and value principles*)
abbreviation utility::cv  ("UTILITY_")  where "UTILITYˣ  ≡ ⦃UTILITY  x⦄"
abbreviation security::cv ("SECURITY_") where "SECURITYˣ ≡ ⦃SECURITY x⦄"
abbreviation equality::cv ("EQUALITY_") where "EQUALITYˣ ≡ ⦃EQUALITY x⦄"
abbreviation freedom::cv  ("FREEDOM_")  where "FREEDOMˣ  ≡ ⦃FREEDOM  x⦄"
abbreviation stab::cv ("STAB_") where "STABˣ ≡ ⦃SECURITY x, UTILITY  x⦄"
abbreviation effi::cv ("EFFI_") where "EFFIˣ ≡ ⦃UTILITY  x, SECURITY x⦄"
abbreviation gain::cv ("GAIN_") where "GAINˣ ≡ ⦃UTILITY  x, FREEDOM  x⦄"
abbreviation will::cv ("WILL_") where "WILLˣ ≡ ⦃FREEDOM  x, UTILITY  x⦄"
abbreviation resp::cv ("RESP_") where "RESPˣ ≡ ⦃FREEDOM  x, EQUALITY x⦄"
abbreviation fair::cv ("FAIR_") where "FAIRˣ ≡ ⦃EQUALITY x, FREEDOM  x⦄"
abbreviation equi::cv ("EQUI_") where "EQUIˣ ≡ ⦃EQUALITY x, SECURITY x⦄"
abbreviation reli::cv ("RELI_") where "RELIˣ ≡ ⦃SECURITY x, EQUALITY x⦄"
```

After defining legal (value) principles as combinations (in this case, sets[28]) of basic values (with respect to a legal party), we need to relate them to propositions (sets of worlds/states) in our logic $\mathcal{PL}$. For this, we employ the *derivation operators* introduced in Section 6, whereby each value principle (set of basic values) becomes associated with a proposition (set of worlds) by means of the operator $\downarrow$ (conversely for $\uparrow$). We encode this by defining the corresponding *incidence* relation, or, equivalently, a function $\mathcal{I}_{\iota\to v}$ mapping worlds/states (type ι) to sets of basic values (type $v = (c)\,\texttt{VAL} \to o$). We define $\downarrow_{v\to\sigma}$ so that, given some set of basic values V_v, $V{\downarrow_\sigma}$ denotes the set of all worlds that are $\mathcal{I}$ related to every value in V (analogously for $\uparrow_{\sigma\to v}$). The modelling in the *Isabelle/HOL* proof assistant is as follows.

```
(**Value Theory*)
consts Irel::"ι⇒v" ("ℐ") (*incidence relation worlds-values*)
 (*derivation operators (cf. theory of "formal concept analysis") *)
abbreviation intent::"σ⇒v" ("_↑") where "W↑ ≡ λv. ∀x. W x ⟶ ℐ x v"
abbreviation extent::"v⇒σ" ("_↓") where "V↓ ≡ λw. ∀x. V x ⟶ ℐ w x"
abbreviation extent_brkt ("[_]") where "[V] ≡ V↓" (*alternative notation*)
```

Thus, we can intuitively read the proposition (set of worlds) denoted by $\mathrm{STAB}^p{\downarrow}$ as (those worlds in which) “the legal principle of STABility is observed with respect to the plaintiff”. For convenience, we introduce square brackets ($[\cdot]$) as an alternative notation to $\downarrow$ postfixing in our DSL, so we have $[V] = V{\downarrow}$.

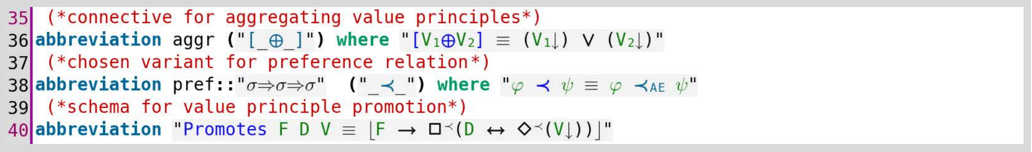
Now, our concrete choice of an aggregation operator for values (out of the two options presented in Section 6.2) is $\oplus_{(2)}$, which thus becomes encoded in HOL as:

$$A_v \oplus_{v\to v\to\sigma} B_v := \ (A\downarrow)_\sigma \vee (B\downarrow)_\sigma$$

Analogously, the chosen preference relation ($\prec$) is the variant $\prec_{AE}$ (i.e., $\prec_{(2)}$ from the candidate modelling options discussed in Section 6), which, recalling Section 5.1, becomes equivalently encoded as any of the following:

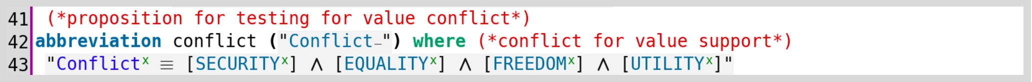
$$\varphi_\sigma \prec_{\sigma\to\sigma\to\sigma} \psi_\sigma := \forall s_i\ \varphi\, s \rightarrow (\exists t_i\ \psi\, t \wedge s \prec t)$$
$$\varphi_\sigma \prec_{\sigma\to\sigma\to\sigma} \psi_\sigma := \mathbf{A}_{\sigma\to\sigma}(\varphi \rightarrow \Diamond^{\prec}_{\sigma\to\sigma}\psi)$$

In a similar fashion, we encode in HOL the value-logical predicate *Promotes* as introduced in the previous subsection Section 6.2. The corresponding *Isabelle/HOL* encoding is shown below.



```
 (*connective for aggregating value principles*)
abbreviation aggr ("[_⊕_]") where "[V₁⊕V₂] ≡ (V₁↓) ∨ (V₂↓)"
 (*chosen variant for preference relation*)
abbreviation pref::"σ⇒σ⇒σ"  ("_≺_") where "φ ≺ ψ ≡ φ ≺AE ψ"
 (*schema for value principle promotion*)
abbreviation "Promotes F D V ≡ ⌊F → □≺(D ↔ ◇≺(V↓))⌋"
```

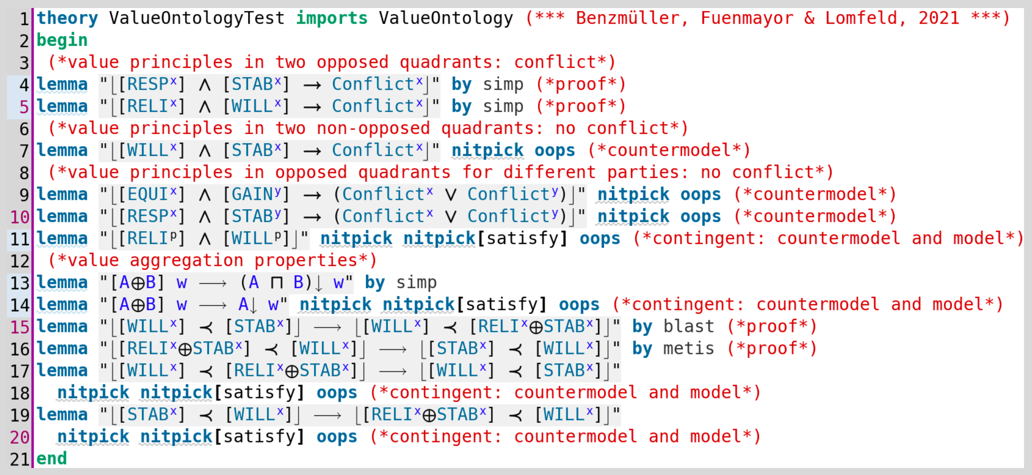
We have similarly encoded the proposition *Conflict* in HOL.



```
 (*proposition for testing for value conflict*)
abbreviation conflict ("Conflict_") where (*conflict for value support*)
 "Conflictˣ ≡ [SECURITYˣ] ∧ [EQUALITYˣ] ∧ [FREEDOMˣ] ∧ [UTILITYˣ]"
```

6.4. Formally Verifying the Adequacy of DSL

In this subsection, we put our HOL-based legal DSL to the test by employing the automated tools integrated into *Isabelle/HOL*. In this process, the *discoursive grammar*, as well as the continuous feedback by our legal domain expert (Lomfeld), served the role of a requirements specification for the formal verification of the adequacy of our modelling. We briefly discuss some of the conducted tests as shown in Figure 6; further tests are presented in Figure A9 in Appendix A.2 and in Benzmüller and Fuenmayor [14].



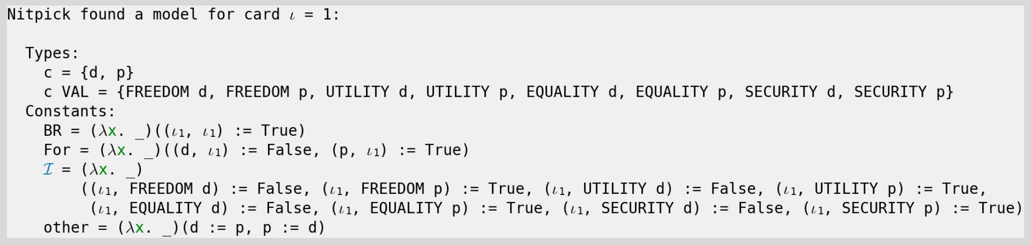
```
 1 theory ValueOntologyTest imports ValueOntology (*** Benzmüller, Fuenmayor & Lomfeld, 2021 ***)
 2 begin
 3  (*value principles in two opposed quadrants: conflict*)
 4 lemma "⌊[RESPˣ] ∧ [STABˣ] → Conflictˣ⌋" by simp (*proof*)
 5 lemma "⌊[RELIˣ] ∧ [WILLˣ] → Conflictˣ⌋" by simp (*proof*)
 6  (*value principles in two non-opposed quadrants: no conflict*)
 7 lemma "⌊[WILLˣ] ∧ [STABˣ] → Conflictˣ⌋" nitpick oops (*countermodel*)
 8  (*value principles in opposed quadrants for different parties: no conflict*)
 9 lemma "⌊[EQUIˣ] ∧ [GAINʸ] → (Conflictˣ ∨ Conflictʸ)⌋" nitpick oops (*countermodel*)
10 lemma "⌊[RESPˣ] ∧ [STABʸ] → (Conflictˣ ∨ Conflictʸ)⌋" nitpick oops (*countermodel*)
11 lemma "⌊[RELIᵖ] ∧ [WILLᵖ]⌋" nitpick nitpick[satisfy] oops (*contingent: countermodel and model*)
12  (*value aggregation properties*)
13 lemma "[A⊕B] w ⟶ (A ⊓ B)↓ w" by simp
14 lemma "[A⊕B] w ⟶ A↓ w" nitpick nitpick[satisfy] oops (*contingent: countermodel and model*)
15 lemma "⌊[WILLˣ] ≺ [STABˣ]⌋ ⟶ ⌊[WILLˣ] ≺ [RELIˣ⊕STABˣ]⌋" by blast (*proof*)
16 lemma "⌊[RELIˣ⊕STABˣ] ≺ [WILLˣ]⌋ ⟶ ⌊[STABˣ] ≺ [WILLˣ]⌋" by metis (*proof*)
17 lemma "⌊[WILLˣ] ≺ [RELIˣ⊕STABˣ]⌋ ⟶ ⌊[WILLˣ] ≺ [STABˣ]⌋"
18   nitpick nitpick[satisfy] oops (*contingent: countermodel and model*)
19 lemma "⌊[STABˣ] ≺ [WILLˣ]⌋ ⟶ ⌊[RELIˣ⊕STABˣ] ≺ [WILLˣ]⌋"
20   nitpick nitpick[satisfy] oops (*contingent: countermodel and model*)
21 end
```

Figure 6. Verifying the DSL.

In accordance with the dialectical interpretation of the *discoursive grammar* (recall Figure 2 in Section 2), our modelling foresees that observing values (with respect to the same party) from two opposing value quadrants, say RESP and STAB, or RELI and WILL, entails a value conflict; theorem provers quickly confirm this as shown in Figure 6 (Lines 4–5).

Moreover, observing values from two non-opposed quadrants, such as WILL and STAB (Line 7), should not imply any conflict: the model finder *Nitpick*[29] computes and reports a countermodel (not shown here) to the stated conjecture. A value conflict is also not implied if values from opposing quadrants are observed with respect to different parties (Lines 9–10).

Note that the notion of value conflict has deliberately not been aligned with logical inconsistency, neither in the object logic $\mathcal{PL}$ nor in the meta-logic HOL. Instead, an explicit, legal party-dependent notion of conflict is introduced as an additional predicate. This way, we can represent conflict situations in which, for instance, RELI and WILL (being conflicting values, see Line 5 in Figure 6) are observed with respect to the plaintiff (p), without leading to a logical inconsistency in *Isabelle/HOL* (thus avoiding 'explosion'). This also has the technical advantage that value conflicts can be explicitly analysed and reported by the model finder *Nitpick*, which would otherwise just report that there are no satisfying models. In Line 11 of Figure 6, for example, *Nitpick* is called simultaneously in both modes in order to confirm the contingency of the statement; as expected, both a model (cf. Figure 7) and countermodel (not displayed here) for the statement are returned. This value conflict can also be spotted by inspecting the satisfying models generated by *Nitpick*. One such model is depicted in Figure 7, where it is shown that (in the given possible world ι_1) all of the basic values (EQUALITY, SECURITY, UTILITY, and FREEDOM) are simultaneously observed with respect to p, which implies a value conflict according to our definition. For further illustrations of such models (with and without value conflict), we refer to the tests reported in Figure A9 in Appendix A.2.



```
Nitpick found a model for card ι = 1:

  Types:
    c = {d, p}
    c VAL = {FREEDOM d, FREEDOM p, UTILITY d, UTILITY p, EQUALITY d, EQUALITY p, SECURITY d, SECURITY p}
  Constants:
    BR = (λx. _)((ι1, ι1) := True)
    For = (λx. _)((d, ι1) := False, (p, ι1) := True)
    𝓘 = (λx. _)
        ((ι1, FREEDOM d) := False, (ι1, FREEDOM p) := True, (ι1, UTILITY d) := False, (ι1, UTILITY p) := True,
         (ι1, EQUALITY d) := False, (ι1, EQUALITY p) := True, (ι1, SECURITY d) := False, (ι1, SECURITY p) := True)
    other = (λx. _)(d := p, p := d)
```

Figure 7. Satisfying model for the statement in Line 11 of Figure 6.

Such model structures as computed by *Nitpick* are ideally *communicated to* (and *inspected with*) domain experts (Lomfeld in our case) early on and checked for plausibility, which, in the case of issues, might trigger adaptions to the axioms and definitions. Such a process may require several cycles until arriving at a state of *reflective equilibrium* (recall the discussion from Section 3) and, as a useful side effect, it conveniently fosters cross-disciplinary mutual understanding.

Further tests in Figure 6 (Lines 13–20) assess the behaviour of the aggregation operator $\oplus$ by itself, and also in combination with value preferences. For example, we test for a correct behaviour when 'strengthening' the right-hand side: if STAB is preferred over WILL, then STAB combined with, say, RELI is also preferred over WILL alone (Line 15). Similar tests are conducted for the 'weakening' of the left-hand side[30].

7. Applications (L3)—Assessment of Legal Cases

In this section, we provide a concrete illustration of our reasoning framework by formally encoding and assessing two classic common law property cases concerning the appropriation of wild animals ("wild animal cases"): Pierson v. Post, and Conti v. ASPCA[31].

Before starting with the analysis, a word is in order about the support of our work by the tools *Sledgehammer* (Blanchette et al. [61,98]) and *Nitpick* (Blanchette and Nipkow [95]) in *Isabelle/HOL*. The ATP systems integrated via *Sledgehammer* in *Isabelle/HOL* include higher-order ATP systems, first-order ATP systems, and SMT (satisfiability modulo theories) solvers, and many of these systems in turn use efficient SAT solver technology internally. Indeed, proof automation with *Sledgehammer* and (counter) model finding with *Nitpick* were invaluable in supporting our exploratory modelling approach at various levels. These tools were very responsive in automatically proving (*Sledgehammer*), disproving (*Nitpick*), or showing consistency by providing a model (*Nitpick*). In the first case, references to the required axioms and lemmas were returned (which can be seen as a kind of abduction), and in the case of models and counter-models, they often proved to be very readable and intuitive. In this section, we highlight some explicit use cases of *Sledgehammer* and *Nitpick*. They have been similarly applied at all levels as mentioned before.

We have split our analysis in layer L3 into two 'sub-layers' in order to highlight the separation between general legal and world knowledge (legal concepts and norms), from its 'application' to relevant facts in the process of deciding a case (factual/contextual knowledge). We shall first address the modelling of some background legal and world knowledge in Section 7.1 as minimally required in order to formulate each of our legal cases in the form of a logical *Isabelle/HOL* theory (cf. Section 7.2).

7.1. General Legal and World Knowledge

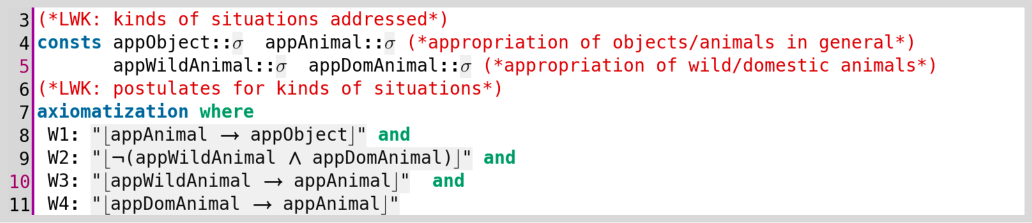
The realistic modelling of concrete legal cases requires further legal and world knowledge (LWK) to be taken into account. LWK is typically modelled in so-called "upper" and "domain" ontologies. The question about which particular notion belongs to which category is difficult, and apparently there is no generally agreed answer in the literature. Anyhow, we introduce only a small and monolithic exemplary logical *Isabelle/HOL* theory[32], called "GeneralKnowledge", with a minimal amount of axioms and definitions as required to encode our legal cases. This LWK example includes a small excerpt of a much simplified "animal appropriation taxonomy", where we associate "animal appropriation" (kinds of) situations with the value preferences they imply (i.e., conditional preference relations as discussed in Sections 2 and 6).

In a more realistic setting, this knowledge base would be further split and structured similarly to other legal or general ontologies, e.g., in the *Semantic Web* (Casanovas et al. [8], Hoekstra et al. [9]). Note, however, that the expressiveness in our approach, unlike in many other legal ontologies or taxonomies, is by no means limited to definite underlying (but fixed) logical language foundations. We could thus easily decide for a more realistic modelling, e.g., avoiding simplifying propositional abstractions. For instance, the proposition "appWildAnimal", representing the appropriation of one or more wild animals, can anytime be replaced by a more complex formula (featuring, for example, quantifiers, modalities, and conditionals; see Section 5.4).

Next steps include interrelating notions introduced in our *Isabelle/HOL theory* "GeneralKnowledge" with values and value preferences as introduced in the previous sections. It is here where the preference relations and modal operators of $\mathcal{PL}$ as well as the notions introduced in our legal DSL are most useful. Remember that, at a later point and in line with the LogiKEy methodology, we may in fact exchange $\mathcal{PL}$ by an alternative choice of an object logic; or, on top of it, we may further modify our legal DSL, e.g., we might choose and assess alternative candidates for our connectives $\prec$ and $\oplus$. Moreover, we may want to replace material implication $\rightarrow$ by a conditional implication to better support defeasible legal reasoning[33].

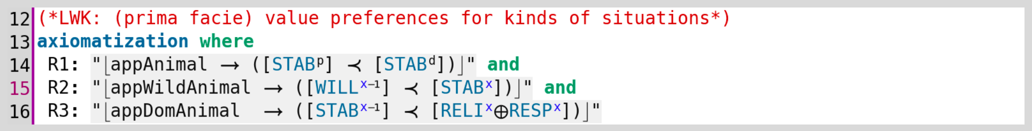
We now briefly outline the *Isabelle/HOL* encoding of our example LWK; see Figure A10 in Appendix A.3 for the full details.

First, some non-logical constants that stand for kinds of legally relevant situations (here, of appropriation) are introduced, and their meaning is constrained by some postulates:



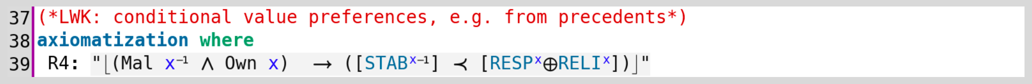
```
(*LWK: kinds of situations addressed*)
consts appObject::σ  appAnimal::σ (*appropriation of objects/animals in general*)
       appWildAnimal::σ  appDomAnimal::σ (*appropriation of wild/domestic animals*)
(*LWK: postulates for kinds of situations*)
axiomatization where
 W1: "⌊appAnimal  → appObject⌋" and
 W2: "⌊¬(appWildAnimal ∧ appDomAnimal)⌋" and
 W3: "⌊appWildAnimal → appAnimal⌋"  and
 W4: "⌊appDomAnimal  → appAnimal⌋"
```

Then, the ‘default’[34] legal rules for several situations (here, the appropriation of animals) are formulated as conditional preference relations.



```
(*LWK: (prima facie) value preferences for kinds of situations*)
axiomatization where
 R1: "⌊appAnimal  → ([STABᵖ] ≺ [STABᵈ])⌋" and
 R2: "⌊appWildAnimal → ([WILLˣ⁻¹] ≺ [STABˣ])⌋" and
 R3: "⌊appDomAnimal   → ([STABˣ⁻¹] ≺ [RELIˣ⊕RESPˣ])⌋"
```

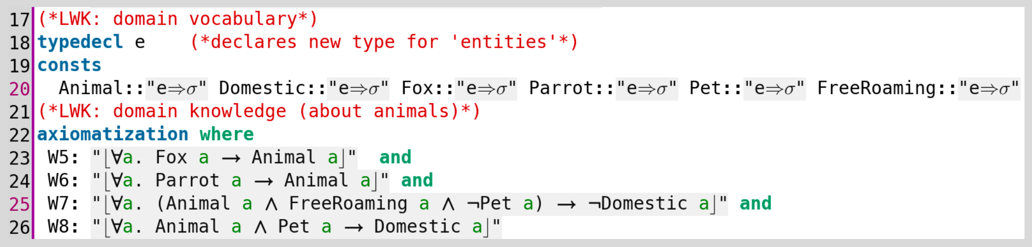
For example, rule R2 could be read as “In a wild-animals-appropriation kind of situation, observing STABility with respect to a party (say, the plaintiff) is preferred over observing WILL with respect to the other party (defendant)”. If there is no more specific legal rule from a precedent or a codified statute, then these ‘default’ preference relations determine the result. Of course, this default is not arbitrary but is itself an implicit normative setting of the existing legal statutes or cases. Moreover, we can have rules conditioned on more concrete legal *factors*[35]. As a didactic example, the legal rule R4 states that the *own*ership (say, the plaintiff’s) of the land on which the appropriation took place, together with the fact that the opposing party (defendant) acted out of *mal*ice implies a value preference of *reli*ance (in ownership) and *resp*onsibility (for malevolence) over *stab*ility (induced by possession as an obvious external sign of appropriation). This rule has been chosen to reflect the famous common law precedent of Keeble v. Hickeringill (1704, 103 ER 1127; cf. also Berman and Hafner [6], Bench-Capon [96]).



```
(*LWK: conditional value preferences, e.g. from precedents*)
axiomatization where
 R4: "⌊(Mal x⁻¹ ∧ Own x)  → ([STABˣ⁻¹] ≺ [RESPˣ⊕RELIˣ])⌋"
```

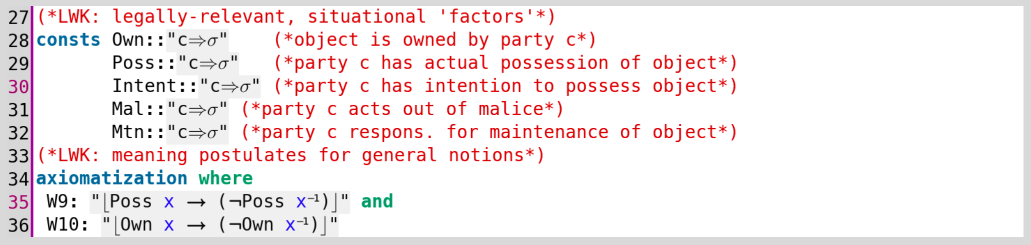
As already discussed, for ease of illustration, terms like “appWildAnimal” are modelled here as simple propositional constants. In practice, however, they may later be replaced, or logically implied, by a more realistic modelling of the relevant situational facts, utilising suitably complex (even quantified; cf. Section 5.4) formulas depicting states of affairs to some desired level of granularity.

For the sake of modelling the appropriation of objects, we have introduced an additional base type in our meta-logic HOL (recall Section 4). The type *e* (for ‘entities’) can be employed for terms denoting individuals (things, animals, etc.) when modelling legally relevant situations. Some simple vocabulary and taxonomic relationships (here, for wild and domestic animals) are specified to illustrate this.



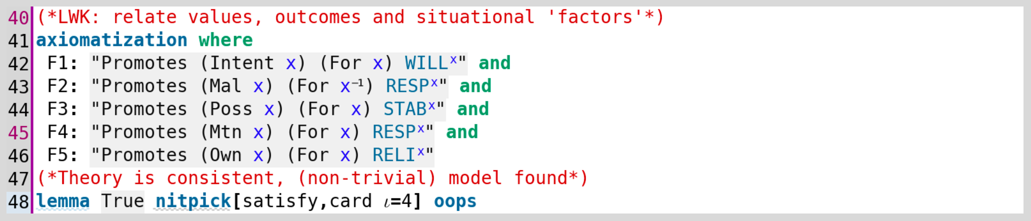
```
(*LWK: domain vocabulary*)
typedecl e    (*declares new type for 'entities'*)
consts
  Animal::"e⇒σ" Domestic::"e⇒σ" Fox::"e⇒σ" Parrot::"e⇒σ" Pet::"e⇒σ" FreeRoaming::"e⇒σ"
(*LWK: domain knowledge (about animals)*)
axiomatization where
 W5: "⌊∀a. Fox a → Animal a⌋"  and
 W6: "⌊∀a. Parrot a → Animal a⌋" and
 W7: "⌊∀a. (Animal a ∧ FreeRoaming a ∧ ¬Pet a) → ¬Domestic a⌋" and
 W8: "⌊∀a. Animal a ∧ Pet a → Domestic a⌋"
```

As mentioned before, we have introduced some convenient legal *factors* into our example LWK to allow for the encoding of legal knowledge originating from precedents or statutes at a more abstract level. In our approach, these factors are to be logically implied (as deductive arguments) from the concrete facts of the case (as exemplified in Appendix A.4 below). Observe that our framework also allows us to introduce definitions for those factors for which clear legal specifications exist, such as property or possession. At the present stage, we will provide some simple postulates constraining the factors' interpretation.



```
(*LWK: legally-relevant, situational 'factors'*)
consts Own::"c⇒σ"    (*object is owned by party c*)
       Poss::"c⇒σ"   (*party c has actual possession of object*)
       Intent::"c⇒σ" (*party c has intention to possess object*)
       Mal::"c⇒σ" (*party c acts out of malice*)
       Mtn::"c⇒σ" (*party c respons. for maintenance of object*)
(*LWK: meaning postulates for general notions*)
axiomatization where
 W9: "⌊Poss x ⟶ (¬Poss x⁻¹)⌋" and
 W10: "⌊Own x ⟶ (¬Own x⁻¹)⌋"
```

Recalling Section 6, we relate the introduced legal factors (and relevant situational facts) to value principles and outcomes by means of the *Promotes* predicate[36]:



```
(*LWK: relate values, outcomes and situational 'factors'*)
axiomatization where
 F1: "Promotes (Intent x) (For x) WILLˣ" and
 F2: "Promotes (Mal x) (For x⁻¹) RESPˣ" and
 F3: "Promotes (Poss x) (For x) STABˣ" and
 F4: "Promotes (Mtn x) (For x) RESPˣ" and
 F5: "Promotes (Own x) (For x) RELIˣ"
(*Theory is consistent, (non-trivial) model found*)
lemma True nitpick[satisfy,card ι=4] oops
```

Finally, the consistency of all axioms and rules provided is confirmed by *Nitpick*.

7.2. Pierson v. Post

This famous legal case can be succinctly described as follows (Bench-Capon et al. [25], Gordon and Walton [97]):

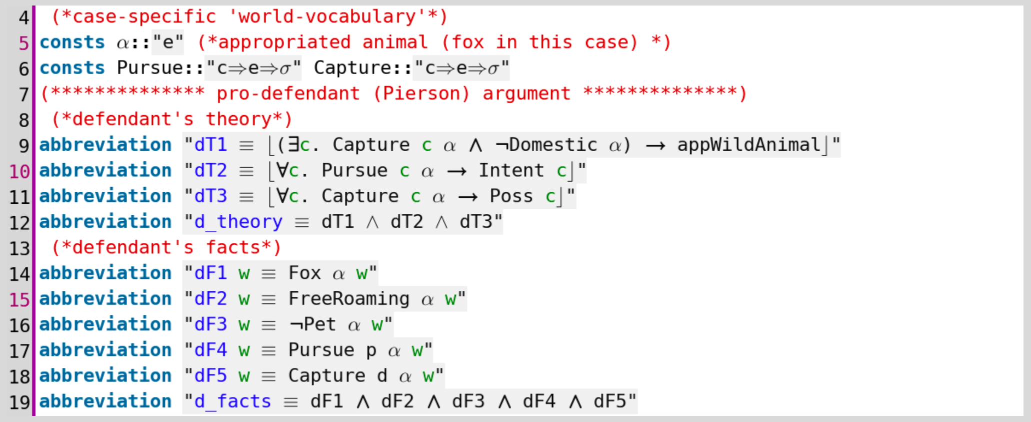
Pierson killed and carried off a fox which Post already was hunting with hounds on public land. The Court found for Pierson (1805, 3 Cai R 175).

For the sake of illustration, we will consider in this subsection two modelling scenarios: in the first one, a case is built to favour the defendant (Pierson), and in the second one, a case favouring the plaintiff (Post).

7.2.1. Ruling for Pierson

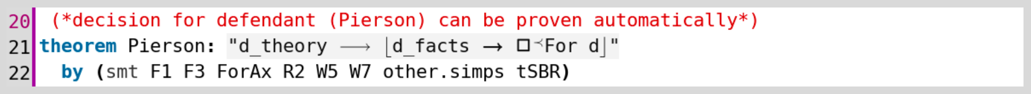
The formal modelling of an argument in favour of Pierson is outlined next[37].

First, we introduce some minimal vocabulary: a constant α of type e (denoting the appropriated animal), and the relations *pursue* and *capture* between the animal and one of the parties (of type c). A background (generic) theory as well as the (contingent) case facts as suitably interpreted by Pierson's party are then stipulated.



```
 (*case-specific 'world-vocabulary'*)
consts α::"e" (*appropriated animal (fox in this case) *)
consts Pursue::"c⇒e⇒σ" Capture::"c⇒e⇒σ"
(************** pro-defendant (Pierson) argument **************)
 (*defendant's theory*)
abbreviation "dT1 ≡ ⌊(∃c. Capture c α ∧ ¬Domestic α) ⟶ appWildAnimal⌋"
abbreviation "dT2 ≡ ⌊∀c. Pursue c α ⟶ Intent c⌋"
abbreviation "dT3 ≡ ⌊∀c. Capture c α ⟶ Poss c⌋"
abbreviation "d_theory ≡ dT1 ∧ dT2 ∧ dT3"
 (*defendant's facts*)
abbreviation "dF1 w ≡ Fox α w"
abbreviation "dF2 w ≡ FreeRoaming α w"
abbreviation "dF3 w ≡ ¬Pet α w"
abbreviation "dF4 w ≡ Pursue p α w"
abbreviation "dF5 w ≡ Capture d α w"
abbreviation "d_facts ≡ dF1 ∧ dF2 ∧ dF3 ∧ dF4 ∧ dF5"
```

The aforementioned decision of the court for Pierson was justified by the majority opinion. The essential preference relation in the case is implied in the idea that the appropriation of (free-roaming) wild animals requires actual corporal possession. The manifest corporal link to the possessor creates legal certainty, which is represented by the value STABility and outweighs the mere WILL to possess by the plaintiff; cf. the arguments of classic lawyers cited by the majority opinion: "pursuit alone vests no property" (Justinian), and "corporal possession creates legal certainty" (Pufendorf). According to the *discoursive grammar* in Section 2 (cf. Figure 2), this corresponds to a basic value preference of SECURITY over FREEDOM. We can see that this legal rule R2 as introduced in the previous section (Section 7.1)[38] is indeed employed by *Isabelle/HOL*'s automated tools to prove that, given a suitable defendant's theory, the (contingent) facts imply a decision in favour of Pierson in all 'better' worlds (which we could even give a 'deontic' reading as some sort of *recommendation).*

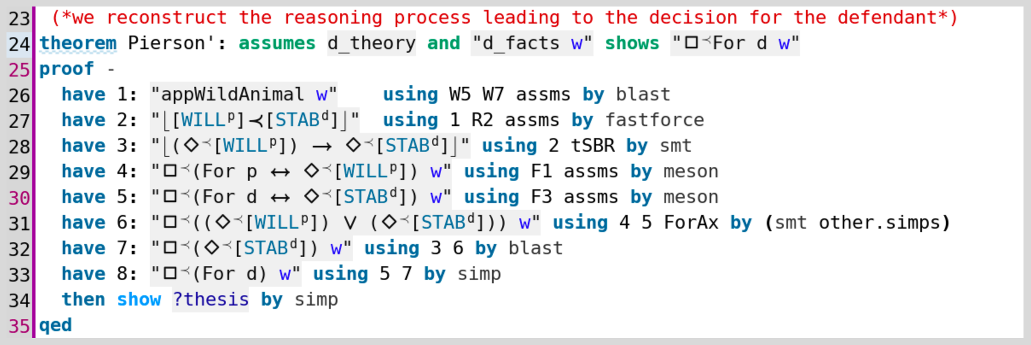


```
 (*decision for defendant (Pierson) can be proven automatically*)
theorem Pierson: "d_theory ⟶ ⌊d_facts → □≺For d⌋"
  by (smt F1 F3 ForAx R2 W5 W7 other.simps tSBR)
```

The previous 'one-liner' proof has indeed been automatically suggested by *Sledgehammer* (Blanchette et al. [61,98]) which we credit, together with the model finder *Nitpick* (Blanchette and Nipkow [95]), for doing the proof heavy-lifting in our work.

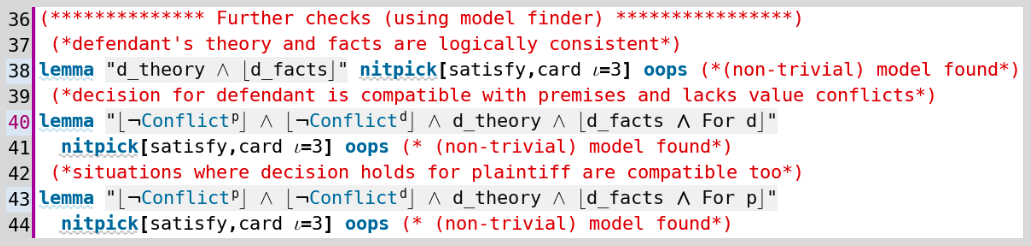
A proof argument in favour of Pierson that uses the same dependencies can also be constructed interactively using *Isabelle*'s human-readable proof language *Isar* (*Isabelle/Isar*; cf. Wenzel [101]). The individual steps of the proof are, this time, formulated with respect to an explicit world/situation parameter w. The argument goes roughly as follows:

1. From Pierson's facts and theory, we infer that in the disputed situation w, a wild animal has been appropriated: appWildAnimal w.
2. In this context, by applying the value preference rule R2, we obtain that observing STAB with respect to Pierson (d) is preferred over observing WILL with respect to Post (p): $\lfloor [\text{WILL}^p] \prec [\text{STAB}^d] \rfloor$.
3. The possibility of observing WILL with respect to Post thus entails the possibility of observing STAB with respect to Pierson: $\lfloor \Diamond^{\prec}[\text{WILL}^p] \longrightarrow \Diamond^{\prec}[\text{STAB}^d] \rfloor$.
4. Moreover, after instantiating the *value promotion* schema F1 (Section 7.1) for Post (p), and acknowledging that his pursuing the animal (Pursue p α) entails his intention to possess (Intent p), we obtain (for the given situation w) a recommendation to 'align' any ruling for Post with the possibility of observing WILL with respect to Post: $\Box^{\prec}(\text{For } p \leftrightarrow \Diamond^{\prec}[\text{WILL}^p])\ w$. Following the interpretation of the *Promotes* predicate given in Section 6, we can read this 'alignment' as involving both a logical entailment (left to right) and a justification (right to left); thus the possibility of observing WILL (with respect to Post) both entails and justifies (as a reason) a legal decision for Post.
5. Analogously, in view of Pierson's (d) capture of the animal (Capture d α), thus having taken possession of it (Poss d), we infer from the instantiation of *value promotion* schema F3 (for Pierson) a recommendation to align a ruling for Pierson with the possibility of observing the value principle STAB with respect to Pierson): $\Box^{\prec}(\text{For } d \leftrightarrow \Diamond^{\prec}[\text{STAB}^d])\ w$
6. From (4) and (5) in combination with the court's duty to find a ruling for one of both parties (axiom *ForAx*) we infer, for the given situation w, that either the possibility of observing WILL with respect to Post or the possibility of observing STAB with respect to Pierson (or both) hold in every 'better' world/situation (thus becoming a recommended condition): $\Box^{\prec}(\Diamond^{\prec}[\text{WILL}^p] \vee \Diamond^{\prec}[\text{STAB}^d])\ w$.
7. From this and (3), we thus obtain that the possibility of observing STAB with respect to Pierson is recommended in the given situation w: $\Box^{\prec}(\Diamond^{\prec}[\text{STAB}^d])\ w$.
8. And this together with (5) finally implies the recommendation to rule in favour of Pierson in the given situation w: $\Box^{\prec}(\text{For } d\ v)$.



```
23  (*we reconstruct the reasoning process leading to the decision for the defendant*)
24 theorem Pierson': assumes d_theory and "d_facts w" shows "□≺For d w"
25 proof -
26   have 1: "appWildAnimal w"    using W5 W7 assms by blast
27   have 2: "⌊[WILLᵖ]≺[STABᵈ]⌋"  using 1 R2 assms by fastforce
28   have 3: "⌊(◇≺[WILLᵖ]) → ◇≺[STABᵈ]⌋" using 2 tSBR by smt
29   have 4: "□≺(For p ↔ ◇≺[WILLᵖ]) w" using F1 assms by meson
30   have 5: "□≺(For d ↔ ◇≺[STABᵈ]) w" using F3 assms by meson
31   have 6: "□≺((◇≺[WILLᵖ]) ∨ (◇≺[STABᵈ])) w" using 4 5 ForAx by (smt other.simps)
32   have 7: "□≺(◇≺[STABᵈ]) w" using 3 6 by blast
33   have 8: "□≺(For d) w" using 5 7 by simp
34   then show ?thesis by simp
35 qed
```

The consistency of the assumed theory and facts (favouring Pierson) together with the other postulates from the previously introduced logical theories "GeneralKnowledge" and "ValueOntology" is verified by generating a (non-trivial) model using *Nitpick* (Line 38). Further tests confirm that the decision for Pierson (and analogously for Post) is compatible with the premises and, moreover, that for neither party, value conflicts are implied.

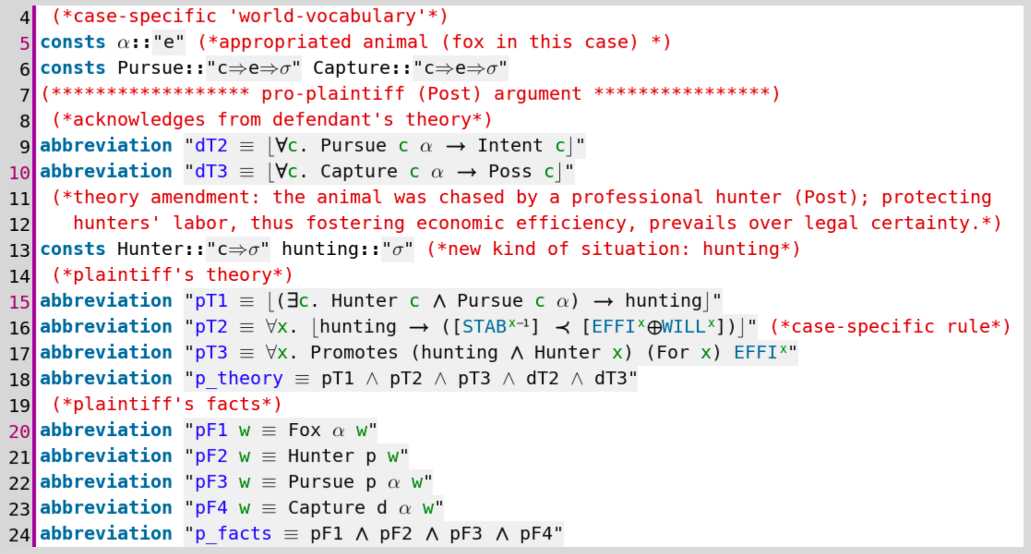


```
36 (************** Further checks (using model finder) ****************)
37  (*defendant's theory and facts are logically consistent*)
38 lemma "d_theory ∧ ⌊d_facts⌋" nitpick[satisfy,card ι=3] oops (*(non-trivial) model found*)
39  (*decision for defendant is compatible with premises and lacks value conflicts*)
40 lemma "⌊¬Conflictᵖ⌋ ∧ ⌊¬Conflictᵈ⌋ ∧ d_theory ∧ ⌊d_facts ∧ For d⌋"
41   nitpick[satisfy,card ι=3] oops (* (non-trivial) model found*)
42  (*situations where decision holds for plaintiff are compatible too*)
43 lemma "⌊¬Conflictᵖ⌋ ∧ ⌊¬Conflictᵈ⌋ ∧ d_theory ∧ ⌊d_facts ∧ For p⌋"
44   nitpick[satisfy,card ι=3] oops (* (non-trivial) model found*)
```

We show next how it is indeed possible to construct a case (theory) suiting Post using our approach.

7.2.2. Ruling for Post

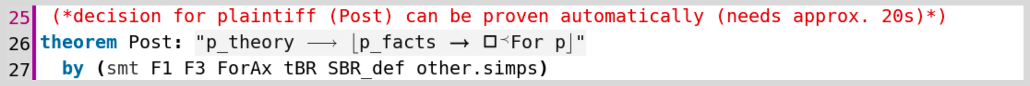
We model a possible counterargument in favour of Post claiming an interpretation (i.e., a *distinction* in case law methodology) in that the animal, being vigorously pursued (with large dogs and hounds) by a professional hunter, is not "free-roaming" anymore but already in (quasi) possession of the hunter. In this interpretation, the value preference $\lfloor[\mathrm{WILL}^p] \prec [\mathrm{STAB}^d]\rfloor$ (for appropriation of wild animals), as in the previous Pierson's argument, is not obtained. Furthermore, Post's party postulates an alternative (suitable) value preference for hunting situations.



```
 4  (*case-specific 'world-vocabulary'*)
 5 consts α::"e" (*appropriated animal (fox in this case) *)
 6 consts Pursue::"c⇒e⇒σ" Capture::"c⇒e⇒σ"
 7 (***************** pro-plaintiff (Post) argument ****************)
 8  (*acknowledges from defendant's theory*)
 9 abbreviation "dT2 ≡ ⌊∀c. Pursue c α → Intent c⌋"
10 abbreviation "dT3 ≡ ⌊∀c. Capture c α → Poss c⌋"
11  (*theory amendment: the animal was chased by a professional hunter (Post); protecting
12    hunters' labor, thus fostering economic efficiency, prevails over legal certainty.*)
13 consts Hunter::"c⇒σ" hunting::"σ" (*new kind of situation: hunting*)
14  (*plaintiff's theory*)
15 abbreviation "pT1 ≡ ⌊(∃c. Hunter c ∧ Pursue c α) → hunting⌋"
16 abbreviation "pT2 ≡ ∀x. ⌊hunting → ([STABˣ⁻¹] ≺ [EFFIˣ⊕WILLˣ])⌋" (*case-specific rule*)
17 abbreviation "pT3 ≡ ∀x. Promotes (hunting ∧ Hunter x) (For x) EFFIˣ"
18 abbreviation "p_theory ≡ pT1 ∧ pT2 ∧ pT3 ∧ dT2 ∧ dT3"
19  (*plaintiff's facts*)
20 abbreviation "pF1 w ≡ Fox α w"
21 abbreviation "pF2 w ≡ Hunter p w"
22 abbreviation "pF3 w ≡ Pursue p α w"
23 abbreviation "pF4 w ≡ Capture d α w"
24 abbreviation "p_facts ≡ pF1 ∧ pF2 ∧ pF3 ∧ pF4"
```

An alternative legal rule (i.e., a possible argument for overruling in case law methodology) is presented (in Line 16 above), entailing a value preference of the value principle combination EFFIciency together with WILL over STABility: $\lfloor[\mathrm{STAB}^d] \prec [\mathrm{EFFI}^p \oplus \mathrm{WILL}^p]\rfloor$. Following the argument put forward by the dissenting opinion in the original case (3 Cai R 175), we might justify this new rule (inverting the initial value preference in the presence of EFFI) by pointing to the alleged public benefit of hunters getting rid of foxes since the latter cause depredations in farms. The hunting of foxes, thus, promotes collective economic utility.

Accepting these modified assumptions, the deductive validity of a decision for Post can in fact be proved and confirmed automatically, again, thanks to *Sledgehammer*:



```
 (*decision for plaintiff (Post) can be proven automatically (needs approx. 20s)*)
theorem Post: "p_theory ⟶ ⌊p_facts → □≺For p⌋"
  by (smt F1 F3 ForAx tBR SBR_def other.simps)
```

Similar to the above, a detailed, interactive proof for the argument in favour of Post is encoded and verified in *Isabelle/Isar*. We have also conducted further tests confirming the consistency of the assumptions and the absence of value conflicts[39].

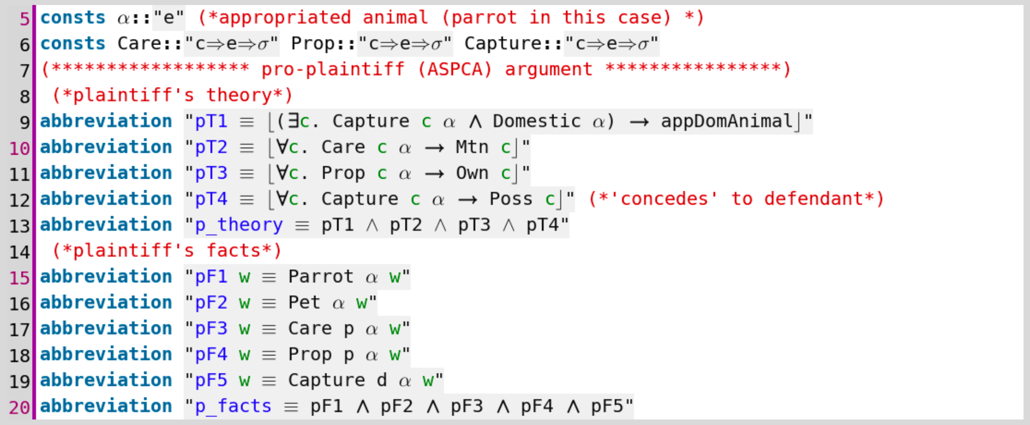
7.3. Conti v. ASPCA

An additional illustrative case study we have modelled in our framework is Conti v. ASPCA (353 NYS 2d 288). In a nutshell is the following (Bench-Capon et al. [25]):

Chester, a parrot owned by the ASPCA, escaped and was recaptured by Conti. The ASPCA found this out and reclaimed Chester from Conti. The court found for ASPCA.

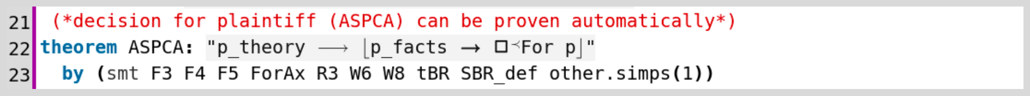
In this case, the court made clear that for domestic animals, the opposite preference relation as the standard in Pierson's case applies. More specifically, it was ruled that for a domestic animal, it is in fact sufficient that the owner did not neglect or stop caring for the animal, i.e., actively give up the responsibility for its maintenance (RESP). This, together with the reliance of ASPCA (RELI) in the parrot's property, outweighs Conti's corporal possession of the animal as a 'stable' external indication (STAB) of property: $\lfloor[\mathrm{STAB}^d] \prec [\mathrm{RELI}^p \oplus \mathrm{RESP}^p]\rfloor$. Observe that a corresponding rule had previously been integrated as R3 into our legal and world knowledge (Section 7.1).

The plaintiff's theory and facts are encoded analogously to the previous case:



```
consts α::"e" (*appropriated animal (parrot in this case) *)
consts Care::"c⇒e⇒σ" Prop::"c⇒e⇒σ" Capture::"c⇒e⇒σ"
(***************** pro-plaintiff (ASPCA) argument ****************)
 (*plaintiff's theory*)
abbreviation "pT1 ≡ ⌊(∃c. Capture c α ∧ Domestic α) → appDomAnimal⌋"
abbreviation "pT2 ≡ ⌊∀c. Care c α → Mtn c⌋"
abbreviation "pT3 ≡ ⌊∀c. Prop c α → Own c⌋"
abbreviation "pT4 ≡ ⌊∀c. Capture c α → Poss c⌋" (*'concedes' to defendant*)
abbreviation "p_theory ≡ pT1 ∧ pT2 ∧ pT3 ∧ pT4"
 (*plaintiff's facts*)
abbreviation "pF1 w ≡ Parrot α w"
abbreviation "pF2 w ≡ Pet α w"
abbreviation "pF3 w ≡ Care p α w"
abbreviation "pF4 w ≡ Prop p α w"
abbreviation "pF5 w ≡ Capture d α w"
abbreviation "p_facts ≡ pF1 ∧ pF2 ∧ pF3 ∧ pF4 ∧ pF5"
```

Accepting these assumptions, the deductive validity of a decision for the plaintiff (ASPCA) can again be proved and confirmed automatically (thanks to *Sledgehammer*):



```
 (*decision for plaintiff (ASPCA) can be proven automatically*)
theorem ASPCA: "p_theory ⟶ ⌊p_facts → □≺For p⌋"
  by (smt F3 F4 F5 ForAx R3 W6 W8 tBR SBR_def other.simps(1))
```

In an analogous manner to Pierson's case, an interactive proof in *Isabelle/Isar* has been encoded and verified, and the consistency of the assumptions and the absence of value conflicts has been confirmed[40].

8. Related and Further Work

Custom software systems for legal case-based reasoning have been developed in the AI and Law community, beginning with the influential HYPO system in the 1980s (Rissland and Ashley [102]); cf. also the survey paper by Bench-Capon [56]. In later years, there has been a gradual shift of interest from rule-based non-monotonic reasoning (e.g., logic programming) towards argumentation-based approaches (see Prakken and Sartor [16] for an overview); however, we are not aware of any other work that uses higher-order theorem proving and proof assistants (the argumentation logic of Krause et al. [103] is an early related effort that is worth mentioning). Another important aspect of our work concerns value-oriented legal reasoning and deliberation, where a considerable amount of work has been presented in AI and Law in response to the challenge posed by Berman and Hafner [6]. Our approach, based mainly on *discoursive grammar* (Lomfeld [4,28]), has also been influenced by some of this work, in particular, by Bench-Capon and Sartor [5], Prakken [51], Bench-Capon [96].

We are currently working towards further refining the modelling of our domain legal theory (*discoursive grammar*) with the aim of providing more expressive (combinations of) object logics at LOGIKEY layer L1. In this regard, it is somehow remarkable that the use of material implication to encode rules has proven sufficient for the illustrative purposes of this paper. However, it is important to note that a more realistic modelling of legal cases must also provide mechanisms to deal with the inevitable emergence of conflicts and contradictions in normative reasoning (overruling, conflict resolution, etc.). In line with the LOGIKEY approach, we are working at introducing conditional connectives in our object logics with the aim of enabling *defeasible* (or *default*) reasoning. Such connectives can be introduced by reusing the modal operators of $\mathcal{PL}$ (recalling the discussion in Section 5.4) or, alternatively, through the shallow semantical embedding (Benzmüller [10]) of a suitable conditional logic in HOL (Benzmüller [81]). Moreover, special kinds of paraconsistent (modal-like) *Logics of Formal Inconsistency* (Carnielli et al. [104]) can also be integrated into our modelling to enable the non-explosive representation of (and recovery from) contradictions by purely object-logical means (cf. Fuenmayor [105] for a related encoding in *Isabelle/HOL*). In a similar vein, we think that some of the recent work that employs expressive deontic logics for value-based legal balancing (e.g., Maranhão and Sartor [27] and the references therein) can be fruitfully integrated in our approach. It is the pluralistic nature of LOGIKEY, realised within a dynamic modelling framework (e.g., *Isabelle/HOL*), that enables and supports such improvements without requiring expensive technical adjustments to the underlying base reasoning technology.

As a broader application scenario, we are currently proposing that ethico-legal value-oriented theories and ontologies should constitute a core ingredient to enable the computation, assessment, and communication of rational justifications and explanations in the future ethico-legal governance of AI (Benzmüller and Lomfeld [106]). Thus, a sound and trustworthy implementation of any legally accountable 'moral machine' requires the development of formal theories and ontologies for the legal domain to guide and interconnect the encoding of concrete regulatory codes and legal cases. Understanding legal reasoning as dialectical practical argumentation, the pluralist interpretation of concrete legal rules arguably requires a complementary ethico-legal value-oriented theory, such as the *discoursive grammar* of justification by Lomfeld [4], which we formally encoded in this paper. In this sense, some first positive evidence has been provided regarding challenges that we have previously identified with respect to the ethical–legal governance of future AI systems (Fuenmayor and Benzmüller [107]). Indeed, it was this broader vision that primarily motivated our work on value-oriented legal reasoning in the first place.

9. Conclusions

We illustrate the application of the LogiKEy knowledge engineering methodology and framework to enable the interdisciplinary collaboration among different specialist roles. In the present case, they are a lawyer and legal philosopher and two computer scientists who joined forces with the aim of formally modelling a value-oriented legal theory (*discoursive grammar* by Lomfeld [4]) for the sake of providing means for computer-automated prediction and assessment of legal case decisions.

From a technical perspective, the core objective of this article has been to demonstrate that the LogiKEy methodology appears indeed suitable for the task of value-oriented legal reasoning. As instantiated in the present work, the LogiKEy methodology builds upon a HOL-encoding of a modal logic of preferences to model a domain-specific theory of value-based legal balancing. In combination with further legal and world knowledge, this theory has been successfully employed for the formal encoding and computer-supported assessment, using the *Isabelle/HOL* system, of illustrative legal cases in property law ("wild animal cases").

It is the flexibility of the multi-layer modelling which is novel in our approach, in combination with a very rich support for automated reasoning in expressive, quantified classical and non-classical logics, thereby rejecting the idea that knowledge representation means should be limited prima facie to decidable logic frameworks, due to complexity or performance considerations[41]. In the LogiKEy approach, the choice of a particular object logic is deliberately left to the knowledge engineer. The range of options varies from well-manageable decidable logics to sophisticated quantified non-classical logics and combinations thereof, depending on what is best suited to handle a particular knowledge representation (and reasoning) task at hand.

From a legal perspective, the reconstruction of legal balancing is, already with classical argumentative tools, a non-trivial task, which is methodologically not yet settled. Here, our work proposed the structuring of legal balancing by means of a dialectical ethico-legal value system (*discoursive grammar*). Legal rules and their various interpretations can thus be represented within a unified yet pluralistic logic of value preferences. The integration of this logic and the value system within the dynamic HOL-based modelling environment allows us to experiment with different forms of interpretation. This enables us not only to find more accurate reconstructions of legal argumentation but also supports the modelling of value-based legal balancing, taking into account notions of value preference, aggregation, promotion and conflict, and also in a manner amenable to computer automation. The modelling of *discoursive grammar* in LogiKEy enabled us to successfully predict (and, to some extent, justify) case outcomes by 'just using logic', employing qualitative value preferences without the necessity to bring in numbers and weights into the model. At the same time, our account could easily be extended to a quantitative model of legal balancing.

From a general perspective, supporting interactive and automated value-oriented legal argumentation on the computer is a non-trivial challenge which we address, for reasons as defended by Bench-Capon [111], with symbolic AI techniques and formal methods. Motivated by recent pleas for *explainable and trustworthy AI*, our primary goal is to work towards the development of ethico-legal governors for future generations of intelligent systems, or more generally, towards some form of legally and ethically *reasonable machines* (Benzmüller and Lomfeld [106]) capable of exchanging rational justifications for the actions they take. While building up a capacity to engage in value-oriented legal argumentation is just one of a multitude of challenges this vision is faced with, it clearly constitutes an important stepping stone towards this ambitious long-term goal.

Author Contributions: Conceptualization, all authors; methodology, C.B. and D.F.; formal modeling, D.F. and C.B.; validation, D.F. and C.B.; writing—original draft preparation, D.F., C.B. and B.L.; writing—review and editing, C.B., D.F. and B.L.; visualization, D.F.; supervision, C.B.; project administration, C.B.; funding acquisition, C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. However, it was intellectually supported by the Fonds National de la Recherche Luxembourg through the projects "Automated Reasoning with Legal Entities (AuReLeE)" (CORE C20/IS/14616644) and "Deontic Logic for Epistemic Rights (DELIGHT)" (OPEN O20/14776480).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is available at https://github.com/cbenzmueller/LogiKEy/tree/master/Preference-Logics/EncodingLegalBalancing (accessed on 12 December 2023)

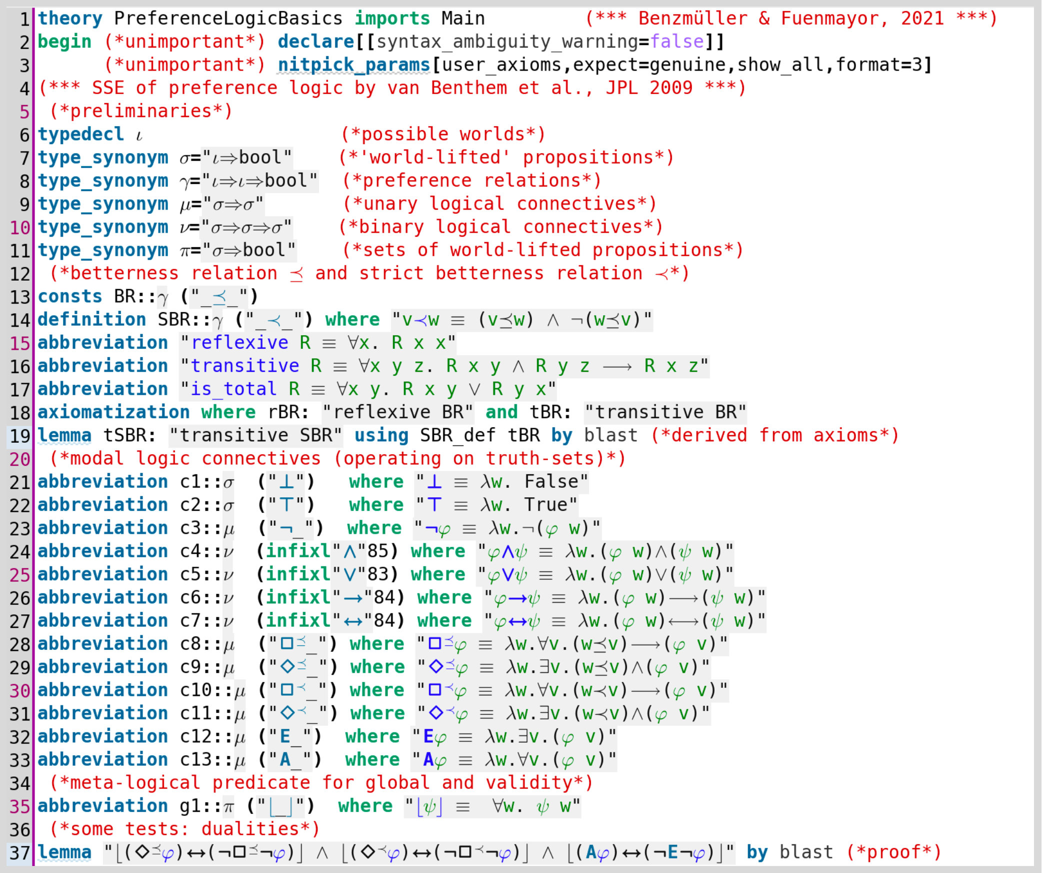
Acknowledgments: We would like to thank the reviewers for their valuable comments. Our thanks also go to the entire LogiKEy team, especially our collaboration partners at the University of Luxembourg.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Appendix —*Isabelle/HOL* Encoding

Appendix A.1. SSE of $\mathcal{PL}$ in HOL

We comment on the implementation of the SSE of $\mathcal{PL}$ in *Isabelle/HOL* as displayed in Figures A1 and A2; see van van Benthem et al. [15] for further details on $\mathcal{PL}$. The defined theory is named "PreferenceLogicBasics", and it relies on base logic HOL, imported here as theory "Main".

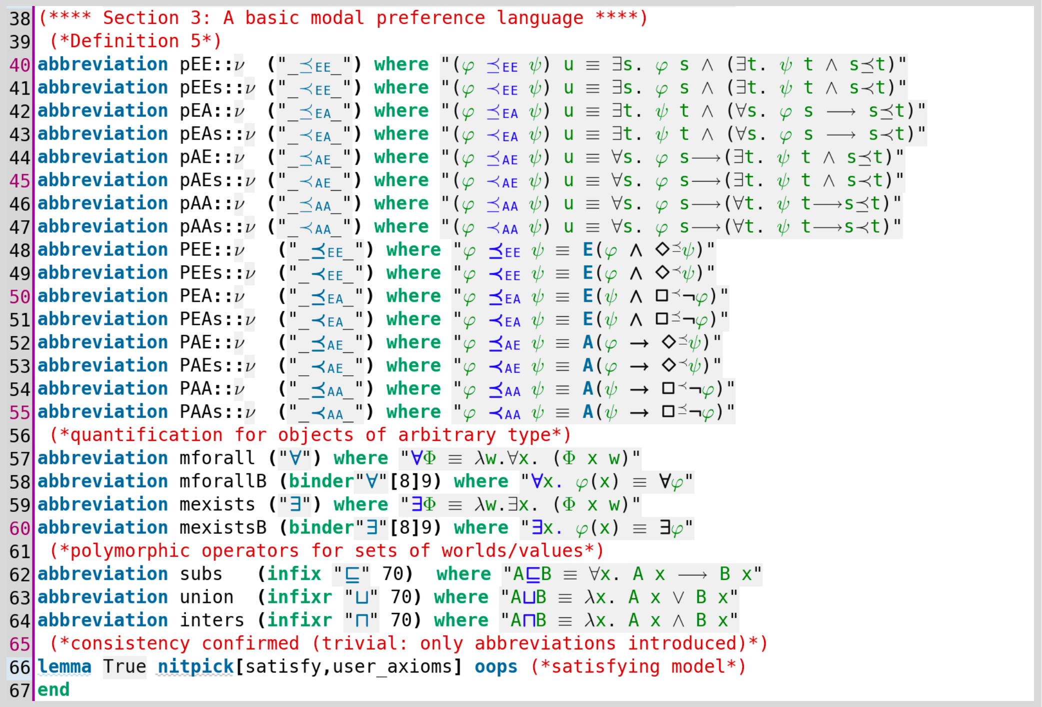


```
 1 theory PreferenceLogicBasics imports Main          (*** Benzmüller & Fuenmayor, 2021 ***)
 2 begin (*unimportant*) declare[[syntax_ambiguity_warning=false]]
 3       (*unimportant*) nitpick_params[user_axioms,expect=genuine,show_all,format=3]
 4 (*** SSE of preference logic by van Benthem et al., JPL 2009 ***)
 5  (*preliminaries*)
 6 typedecl ι                      (*possible worlds*)
 7 type_synonym σ="ι⇒bool"         (*'world-lifted' propositions*)
 8 type_synonym γ="ι⇒ι⇒bool"       (*preference relations*)
 9 type_synonym μ="σ⇒σ"            (*unary logical connectives*)
10 type_synonym ν="σ⇒σ⇒σ"          (*binary logical connectives*)
11 type_synonym π="σ⇒bool"         (*sets of world-lifted propositions*)
12  (*betterness relation ≼ and strict betterness relation ≺*)
13 consts BR::γ ("_≼_")
14 definition SBR::γ ("_≺_") where "v≺w ≡ (v≼w) ∧ ¬(w≼v)"
15 abbreviation "reflexive R ≡ ∀x. R x x"
16 abbreviation "transitive R ≡ ∀x y z. R x y ∧ R y z ⟶ R x z"
17 abbreviation "is_total R ≡ ∀x y. R x y ∨ R y x"
18 axiomatization where rBR: "reflexive BR" and tBR: "transitive BR"
19 lemma tSBR: "transitive SBR" using SBR_def tBR by blast (*derived from axioms*)
20  (*modal logic connectives (operating on truth-sets)*)
21 abbreviation c1::σ  ("⊥")    where "⊥ ≡ λw. False"
22 abbreviation c2::σ  ("⊤")    where "⊤ ≡ λw. True"
23 abbreviation c3::μ  ("¬_")   where "¬φ ≡ λw.¬(φ w)"
24 abbreviation c4::ν  (infixl"∧"85) where "φ∧ψ ≡ λw.(φ w)∧(ψ w)"
25 abbreviation c5::ν  (infixl"∨"83) where "φ∨ψ ≡ λw.(φ w)∨(ψ w)"
26 abbreviation c6::ν  (infixl"→"84) where "φ→ψ ≡ λw.(φ w)⟶(ψ w)"
27 abbreviation c7::ν  (infixl"↔"84) where "φ↔ψ ≡ λw.(φ w)⟷(ψ w)"
28 abbreviation c8::μ  ("□≼_") where "□≼φ ≡ λw.∀v.(w≼v)⟶(φ v)"
29 abbreviation c9::μ  ("◇≼_") where "◇≼φ ≡ λw.∃v.(w≼v)∧(φ v)"
30 abbreviation c10::μ ("□≺_") where "□≺φ ≡ λw.∀v.(w≺v)⟶(φ v)"
31 abbreviation c11::μ ("◇≺_") where "◇≺φ ≡ λw.∃v.(w≺v)∧(φ v)"
32 abbreviation c12::μ ("E_")  where "Eφ ≡ λw.∃v.(φ v)"
33 abbreviation c13::μ ("A_")  where "Aφ ≡ λw.∀v.(φ v)"
34  (*meta-logical predicate for global and validity*)
35 abbreviation g1::π ("⌊_⌋")  where "⌊ψ⌋ ≡  ∀w. ψ w"
36  (*some tests: dualities*)
37 lemma "⌊(◇≼φ)↔(¬□≼¬φ)⌋ ∧ ⌊(◇≺φ)↔(¬□≺¬φ)⌋ ∧ ⌊(Aφ)↔(¬E¬φ)⌋" by blast (*proof*)
```

Figure A1. SSE of $\mathcal{PL}$ (van Benthem et al. [15]) in HOL (continued in Figure A2).

First, a new base type ι is declared (Line 6), denoting the set of possible worlds or states. Subsequently (Lines 7–11), useful type abbreviations are introduced, including the type σ for $\mathcal{PL}$ propositions, which are modelled as predicates on objects of type ι (i.e., as *truth-sets* of worlds/states). A *betterness relation* $\preceq$ and its strict variant $\prec$ are introduced (Lines 13–14), with $\preceq$-accessible worlds interpreted as those that are *at least as good* as the present one. Definitions for relation properties are provided, and it is postulated that $\preceq$ is a preorder, i.e., reflexive and transitive (Lines 15–18).

Subsequently, the σ-type lifted logical connectives of $\mathcal{PL}$ are introduced as abbreviations of λ-terms in the meta-logic HOL (Lines 21–33). The operators $\Box^{\preceq}$ and $\Box^{\prec}$ use $\preceq$ and $\prec$ as guards in their definitions (Lines 28 and 30); analogous for $\Diamond^{\preceq}$ and $\Diamond^{\prec}$. An *universal* modality and its dual are also introduced (Lines 32–33). Moreover, a notion of (global) truth for $\mathcal{PL}$ formulas ψ is defined (Line 35): proposition ψ is globally true, we also say 'valid', if and only if it is true in all worlds.



```
38 (**** Section 3: A basic modal preference language ****)
39  (*Definition 5*)
40 abbreviation pEE::ν  ("_≼EE_") where "(φ ≼EE ψ) u ≡ ∃s. φ s ∧ (∃t. ψ t ∧ s≼t)"
41 abbreviation pEEs::ν ("_≺EE_") where "(φ ≺EE ψ) u ≡ ∃s. φ s ∧ (∃t. ψ t ∧ s≺t)"
42 abbreviation pEA::ν  ("_≼EA_") where "(φ ≼EA ψ) u ≡ ∃t. ψ t ∧ (∀s. φ s ⟶ s≼t)"
43 abbreviation pEAs::ν ("_≺EA_") where "(φ ≺EA ψ) u ≡ ∃t. ψ t ∧ (∀s. φ s ⟶ s≺t)"
44 abbreviation pAE::ν  ("_≼AE_") where "(φ ≼AE ψ) u ≡ ∀s. φ s⟶(∃t. ψ t ∧ s≼t)"
45 abbreviation pAEs::ν ("_≺AE_") where "(φ ≺AE ψ) u ≡ ∀s. φ s⟶(∃t. ψ t ∧ s≺t)"
46 abbreviation pAA::ν  ("_≼AA_") where "(φ ≼AA ψ) u ≡ ∀s. φ s⟶(∀t. ψ t⟶s≼t)"
47 abbreviation pAAs::ν ("_≺AA_") where "(φ ≺AA ψ) u ≡ ∀s. φ s⟶(∀t. ψ t⟶s≺t)"
48 abbreviation PEE::ν   ("_≼EE_") where "φ ≼EE ψ ≡ E(φ ∧ ◇≼ψ)"
49 abbreviation PEEs::ν  ("_≺EE_") where "φ ≺EE ψ ≡ E(φ ∧ ◇≺ψ)"
50 abbreviation PEA::ν   ("_≼EA_") where "φ ≼EA ψ ≡ E(ψ ∧ □≺¬φ)"
51 abbreviation PEAs::ν  ("_≺EA_") where "φ ≺EA ψ ≡ E(ψ ∧ □≼¬φ)"
52 abbreviation PAE::ν   ("_≼AE_") where "φ ≼AE ψ ≡ A(φ → ◇≼ψ)"
53 abbreviation PAEs::ν  ("_≺AE_") where "φ ≺AE ψ ≡ A(φ → ◇≺ψ)"
54 abbreviation PAA::ν   ("_≼AA_") where "φ ≼AA ψ ≡ A(ψ → □≺¬φ)"
55 abbreviation PAAs::ν  ("_≺AA_") where "φ ≺AA ψ ≡ A(ψ → □≼¬φ)"
56  (*quantification for objects of arbitrary type*)
57 abbreviation mforall ("∀") where "∀Φ ≡ λw.∀x. (Φ x w)"
58 abbreviation mforallB (binder"∀"[8]9) where "∀x. φ(x) ≡ ∀φ"
59 abbreviation mexists ("∃") where "∃Φ ≡ λw.∃x. (Φ x w)"
60 abbreviation mexistsB (binder"∃"[8]9) where "∃x. φ(x) ≡ ∃φ"
61  (*polymorphic operators for sets of worlds/values*)
62 abbreviation subs   (infix "⊑" 70)  where "A⊑B ≡ ∀x. A x ⟶ B x"
63 abbreviation union  (infixr "⊔" 70) where "A⊔B ≡ λx. A x ∨ B x"
64 abbreviation inters (infixr "⊓" 70) where "A⊓B ≡ λx. A x ∧ B x"
65  (*consistency confirmed (trivial: only abbreviations introduced)*)
66 lemma True nitpick[satisfy,user_axioms] oops (*satisfying model*)
67 end
```

Figure A2. SSE of $\mathcal{PL}$ (van Benthem et al. [15]) in HOL (continued from Figure A1).

As a first test, some expected dualities of the modal operators are automatically proved (Line 36).

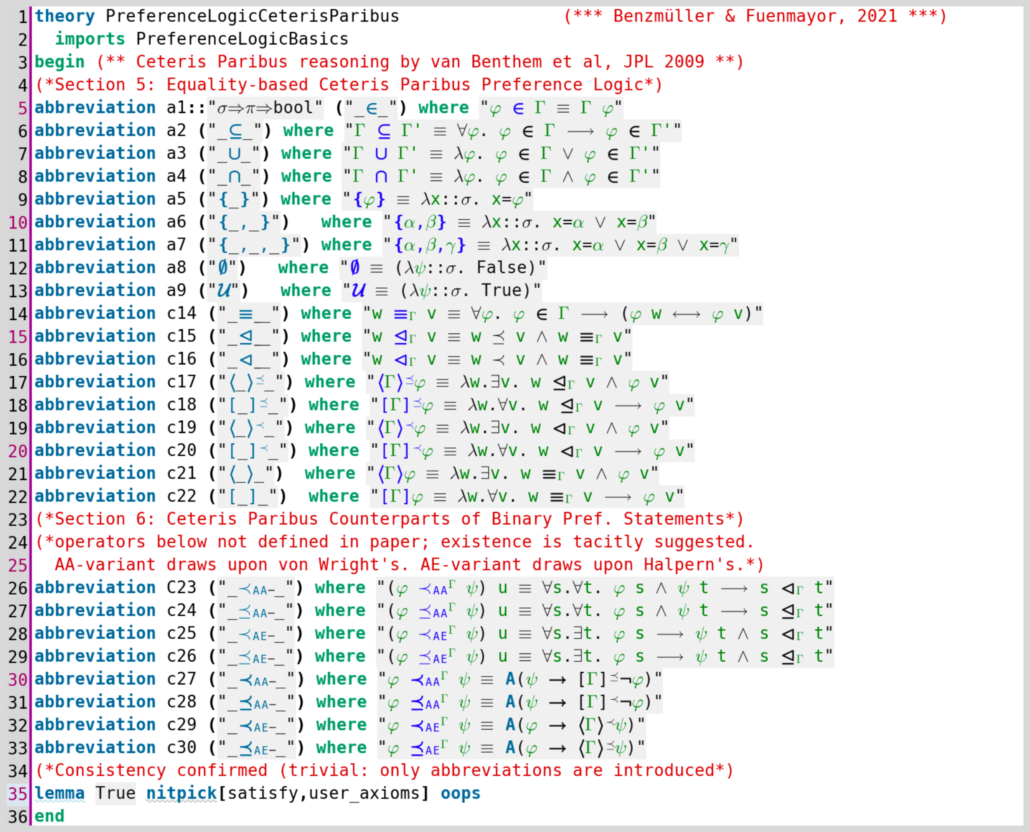
Subsequently, the *betterness* ordering $\preceq$ (respectively, $\prec$) is lifted to a preference relation between $\mathcal{PL}$ propositions (sets of worlds). Eight possible semantic definitions for such preferences are encoded in HOL (Lines 40–47 in Figure A2). The semantic definitions are complemented by eight syntactic definitions of the same binary preferences stated within the object language $\mathcal{PL}$ (Lines 48–55). (ATP systems prove the meta-theoretic correspondences between these semantic and syntactic definitions; cf. Lines 4–12 in Figure A4).

$\mathcal{PL}$ is extended by adding quantifiers (Lines 57–60); cf. (Benzmüller and Paulson [76]) for explanations on the SSE of quantified modal logics. Moreover, useful polymorphic operators for subset, union, and intersection are defined (Lines 62–64).

The model finder *Nitpick* (Blanchette and Nipkow [95]) confirms the consistency of the introduced theory (Line 66) by generating and presenting a model (not shown here), in which the relation satisfies the constraints imposed on it. Thus, the axioms of object logic are simultaneously satisfiable.

To gain practical evidence for the faithfulness of our SSE of $\mathcal{PL}$ in *Isabelle/HOL*, and also to assess the proof automation performance, we have conducted numerous experiments, in which we automatically reconstruct the meta-theoretical results on $\mathcal{PL}$; see Figures A4 and A5.

Extending our SSE of $\mathcal{PL}$ in HOL, some further preference relations for $\mathcal{PL}$ are defined in Figure A3. These additional relations support ceteris paribus reasoning in $\mathcal{PL}$.



```
 1 theory PreferenceLogicCeterisParibus                    (*** Benzmüller & Fuenmayor, 2021 ***)
 2   imports PreferenceLogicBasics
 3 begin (** Ceteris Paribus reasoning by van Benthem et al, JPL 2009 **)
 4 (*Section 5: Equality-based Ceteris Paribus Preference Logic*)
 5 abbreviation a1::"σ⇒π⇒bool" ("_∈_") where "φ ∈ Γ ≡ Γ φ"
 6 abbreviation a2 ("_⊆_") where "Γ ⊆ Γ' ≡ ∀φ. φ ∈ Γ ⟶ φ ∈ Γ'"
 7 abbreviation a3 ("_∪_") where "Γ ∪ Γ' ≡ λφ. φ ∈ Γ ∨ φ ∈ Γ'"
 8 abbreviation a4 ("_∩_") where "Γ ∩ Γ' ≡ λφ. φ ∈ Γ ∧ φ ∈ Γ'"
 9 abbreviation a5 ("{_}") where "{φ} ≡ λx::σ. x=φ"
10 abbreviation a6 ("{_,_}")   where "{α,β} ≡ λx::σ. x=α ∨ x=β"
11 abbreviation a7 ("{_,_,_}") where "{α,β,γ} ≡ λx::σ. x=α ∨ x=β ∨ x=γ"
12 abbreviation a8 ("∅")   where "∅ ≡ (λψ::σ. False)"
13 abbreviation a9 ("𝓤")   where "𝓤 ≡ (λψ::σ. True)"
14 abbreviation c14 ("_≡__") where "w ≡Γ v ≡ ∀φ. φ ∈ Γ ⟶ (φ w ⟷ φ v)"
15 abbreviation c15 ("_⊴__") where "w ⊴Γ v ≡ w ≼ v ∧ w ≡Γ v"
16 abbreviation c16 ("_◁__") where "w ◁Γ v ≡ w ≺ v ∧ w ≡Γ v"
17 abbreviation c17 ("⟨_⟩≼_") where "⟨Γ⟩≼φ ≡ λw.∃v. w ⊴Γ v ∧ φ v"
18 abbreviation c18 ("[_]≼_") where "[Γ]≼φ ≡ λw.∀v. w ⊴Γ v ⟶ φ v"
19 abbreviation c19 ("⟨_⟩≺_") where "⟨Γ⟩≺φ ≡ λw.∃v. w ◁Γ v ∧ φ v"
20 abbreviation c20 ("[_]≺_") where "[Γ]≺φ ≡ λw.∀v. w ◁Γ v ⟶ φ v"
21 abbreviation c21 ("⟨_⟩_")  where "⟨Γ⟩φ ≡ λw.∃v. w ≡Γ v ∧ φ v"
22 abbreviation c22 ("[_]_")  where "[Γ]φ ≡ λw.∀v. w ≡Γ v ⟶ φ v"
23 (*Section 6: Ceteris Paribus Counterparts of Binary Pref. Statements*)
24 (*operators below not defined in paper; existence is tacitly suggested.
25   AA-variant draws upon von Wright's. AE-variant draws upon Halpern's.*)
26 abbreviation C23 ("_≺AA-_") where "(φ ≺AA^Γ ψ) u ≡ ∀s.∀t. φ s ∧ ψ t ⟶ s ◁Γ t"
27 abbreviation c24 ("_≼AA-_") where "(φ ≼AA^Γ ψ) u ≡ ∀s.∀t. φ s ∧ ψ t ⟶ s ⊴Γ t"
28 abbreviation c25 ("_≺AE-_") where "(φ ≺AE^Γ ψ) u ≡ ∀s.∃t. φ s ⟶ ψ t ∧ s ◁Γ t"
29 abbreviation c26 ("_≼AE-_") where "(φ ≼AE^Γ ψ) u ≡ ∀s.∃t. φ s ⟶ ψ t ∧ s ⊴Γ t"
30 abbreviation c27 ("_≺AA-_") where "φ ≺AA^Γ ψ ≡ A(ψ → [Γ]≼¬φ)"
31 abbreviation c28 ("_≼AA-_") where "φ ≼AA^Γ ψ ≡ A(ψ → [Γ]≺¬φ)"
32 abbreviation c29 ("_≺AE-_") where "φ ≺AE^Γ ψ ≡ A(φ → ⟨Γ⟩≺ψ)"
33 abbreviation c30 ("_≼AE-_") where "φ ≼AE^Γ ψ ≡ A(φ → ⟨Γ⟩≼ψ)"
34 (*Consistency confirmed (trivial: only abbreviations are introduced*)
35 lemma True nitpick[satisfy,user_axioms] oops
36 end
```

Figure A3. SSE of $\mathcal{PL}$ (van Benthem et al. [15]) in HOL (continued from Figures A1 and A2).

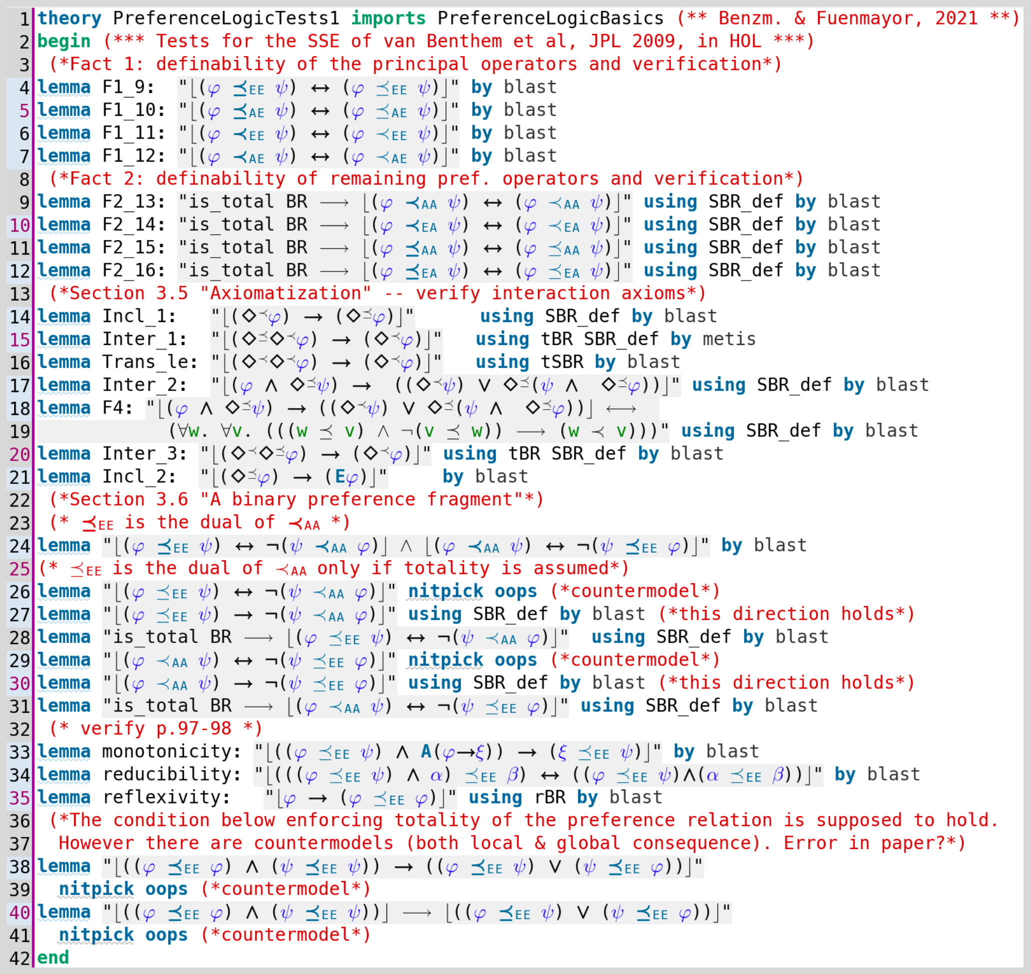
We give some explanations:

Lines 5–13 Useful set theoretic notions are introduced as abbreviations for corresponding λ-terms in HOL.

Lines 14–22 $\mathcal{PL}$ is further extended with (equality-based) ceteris paribus preference relations and modalities; here, Γ represents a set of formulas that are assumed constant between two possible worlds to compare. Hence, our variant can be understood as "these (given) things being equal" preferences. This variant can be used for modelling von Wright's notion of ceteris paribus ("all other things being equal") preferences, eliciting an appropriate Γ by extra-logical means.

Lines 26–33: Except for $\prec^{\Gamma}_{AA}$, the remaining operators we define here were not explicitly defined by van Benthem et al. [15]; however, their existence is tacitly suggested.

Meta-theoretical results on $\mathcal{PL}$ as presented by van Benthem et al. [15] are automatically verified by the reasoning tools in *Isabelle/HOL*; see Figures A3–A7. In fact, we prove all relevant results from (van Benthem et al. [15]). The experiments shown in Figure A4 are briefly commented.



```
 1 theory PreferenceLogicTests1 imports PreferenceLogicBasics (** Benzm. & Fuenmayor, 2021 **)
 2 begin (*** Tests for the SSE of van Benthem et al, JPL 2009, in HOL ***)
 3  (*Fact 1: definability of the principal operators and verification*)
 4 lemma F1_9:  "⌊(φ ≼EE ψ) ↔ (φ ≼EE ψ)⌋" by blast
 5 lemma F1_10: "⌊(φ ≼AE ψ) ↔ (φ ≼AE ψ)⌋" by blast
 6 lemma F1_11: "⌊(φ ≺EE ψ) ↔ (φ ≺EE ψ)⌋" by blast
 7 lemma F1_12: "⌊(φ ≺AE ψ) ↔ (φ ≺AE ψ)⌋" by blast
 8  (*Fact 2: definability of remaining pref. operators and verification*)
 9 lemma F2_13: "is_total BR ⟶ ⌊(φ ≺AA ψ) ↔ (φ ≺AA ψ)⌋" using SBR_def by blast
10 lemma F2_14: "is_total BR ⟶ ⌊(φ ≺EA ψ) ↔ (φ ≺EA ψ)⌋" using SBR_def by blast
11 lemma F2_15: "is_total BR ⟶ ⌊(φ ≼AA ψ) ↔ (φ ≼AA ψ)⌋" using SBR_def by blast
12 lemma F2_16: "is_total BR ⟶ ⌊(φ ≼EA ψ) ↔ (φ ≼EA ψ)⌋" using SBR_def by blast
13  (*Section 3.5 "Axiomatization" -- verify interaction axioms*)
14 lemma Incl_1:   "⌊(◇≺φ) → (◇≼φ)⌋"      using SBR_def by blast
15 lemma Inter_1:  "⌊(◇≼◇≺φ) → (◇≺φ)⌋"   using tBR SBR_def by metis
16 lemma Trans_le: "⌊(◇≺◇≺φ) → (◇≺φ)⌋"   using tSBR by blast
17 lemma Inter_2:  "⌊(φ ∧ ◇≼ψ) →  ((◇≺ψ) ∨ ◇≼(ψ ∧  ◇≼φ))⌋" using SBR_def by blast
18 lemma F4: "⌊(φ ∧ ◇≼ψ) → ((◇≺ψ) ∨ ◇≼(ψ ∧  ◇≼φ))⌋ ⟷
19                (∀w. ∀v. (((w ≼ v) ∧ ¬(v ≼ w)) ⟶ (w ≺ v)))" using SBR_def by blast
20 lemma Inter_3: "⌊(◇≺◇≼φ) → (◇≺φ)⌋" using tBR SBR_def by blast
21 lemma Incl_2:  "⌊(◇≼φ) → (Eφ)⌋"      by blast
22  (*Section 3.6 "A binary preference fragment"*)
23  (* ≼EE is the dual of ≺AA *)
24 lemma "⌊(φ ≼EE ψ) ↔ ¬(ψ ≺AA φ)⌋ ∧ ⌊(φ ≺AA ψ) ↔ ¬(ψ ≼EE φ)⌋" by blast
25 (* ≼EE is the dual of ≺AA only if totality is assumed*)
26 lemma "⌊(φ ≼EE ψ) ↔ ¬(ψ ≺AA φ)⌋" nitpick oops (*countermodel*)
27 lemma "⌊(φ ≼EE ψ) → ¬(ψ ≺AA φ)⌋" using SBR_def by blast (*this direction holds*)
28 lemma "is_total BR ⟶ ⌊(φ ≼EE ψ) ↔ ¬(ψ ≺AA φ)⌋"  using SBR_def by blast
29 lemma "⌊(φ ≺AA ψ) ↔ ¬(ψ ≼EE φ)⌋" nitpick oops (*countermodel*)
30 lemma "⌊(φ ≺AA ψ) → ¬(ψ ≼EE φ)⌋" using SBR_def by blast (*this direction holds*)
31 lemma "is_total BR ⟶ ⌊(φ ≺AA ψ) ↔ ¬(ψ ≼EE φ)⌋" using SBR_def by blast
32  (* verify p.97-98 *)
33 lemma monotonicity: "⌊((φ ≼EE ψ) ∧ A(φ→ξ)) → (ξ ≼EE ψ)⌋" by blast
34 lemma reducibility: "⌊(((φ ≼EE ψ) ∧ α) ≼EE β) ↔ ((φ ≼EE ψ)∧(α ≼EE β))⌋" by blast
35 lemma reflexivity:   "⌊φ → (φ ≼EE φ)⌋" using rBR by blast
36  (*The condition below enforcing totality of the preference relation is supposed to hold.
37   However there are countermodels (both local & global consequence). Error in paper?*)
38 lemma "⌊((φ ≼EE φ) ∧ (ψ ≼EE ψ)) → ((φ ≼EE ψ) ∨ (ψ ≼EE φ))⌋"
39   nitpick oops (*countermodel*)
40 lemma "⌊((φ ≼EE φ) ∧ (ψ ≼EE ψ))⌋ ⟶ ⌊((φ ≼EE ψ) ∨ (ψ ≼EE φ))⌋"
41   nitpick oops (*countermodel*)
42 end
```

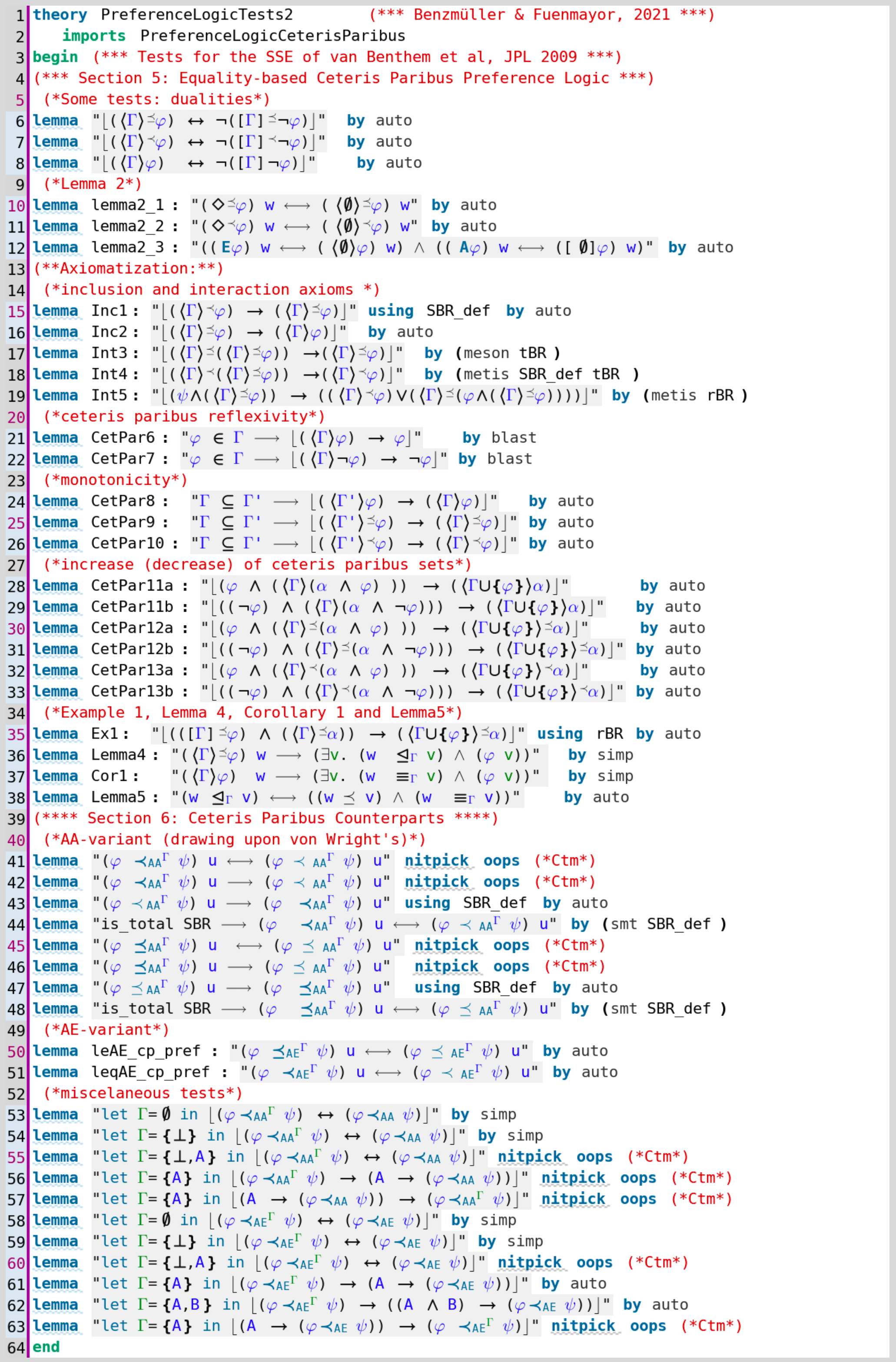
Figure A4. Experiments: testing the meta-theory of $\mathcal{PL}$.

Lines 5–13 Correspondences between the semantically and syntactically defined preference relations are proved.

Lines 15–22 It is proved that (e.g., inclusion and interaction) axioms for $\mathcal{PL}$ follow as theorems in our SSE. This tests the faithfulness of the embedding in one direction.

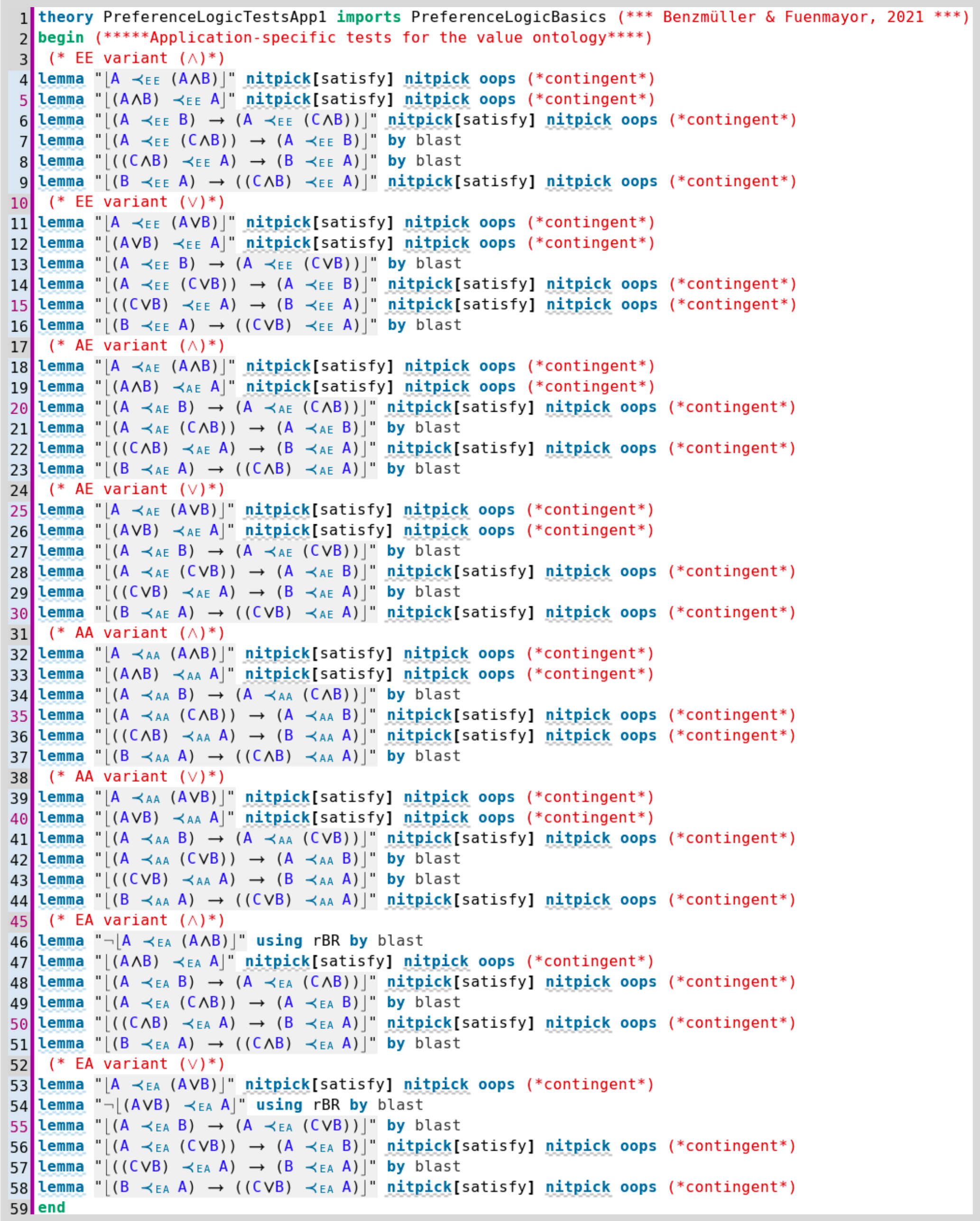
Lines 25–47 We continue the mechanical verification of theorems, and generate countermodels (not displayed here) for non-theorems of $\mathcal{PL}$, thus putting our encoding to the test. Our results coincide with the corresponding ones claimed (and in many cases proved) in van Benthem et al. [15], except for the claims encoded in lines 40-41 and 44-45, where countermodels are reported by *Nitpick*.

Lines 25–47 Some application-specific tests in preparation for the modelling of the legal DSL (including the value theory/ontology) are conducted.



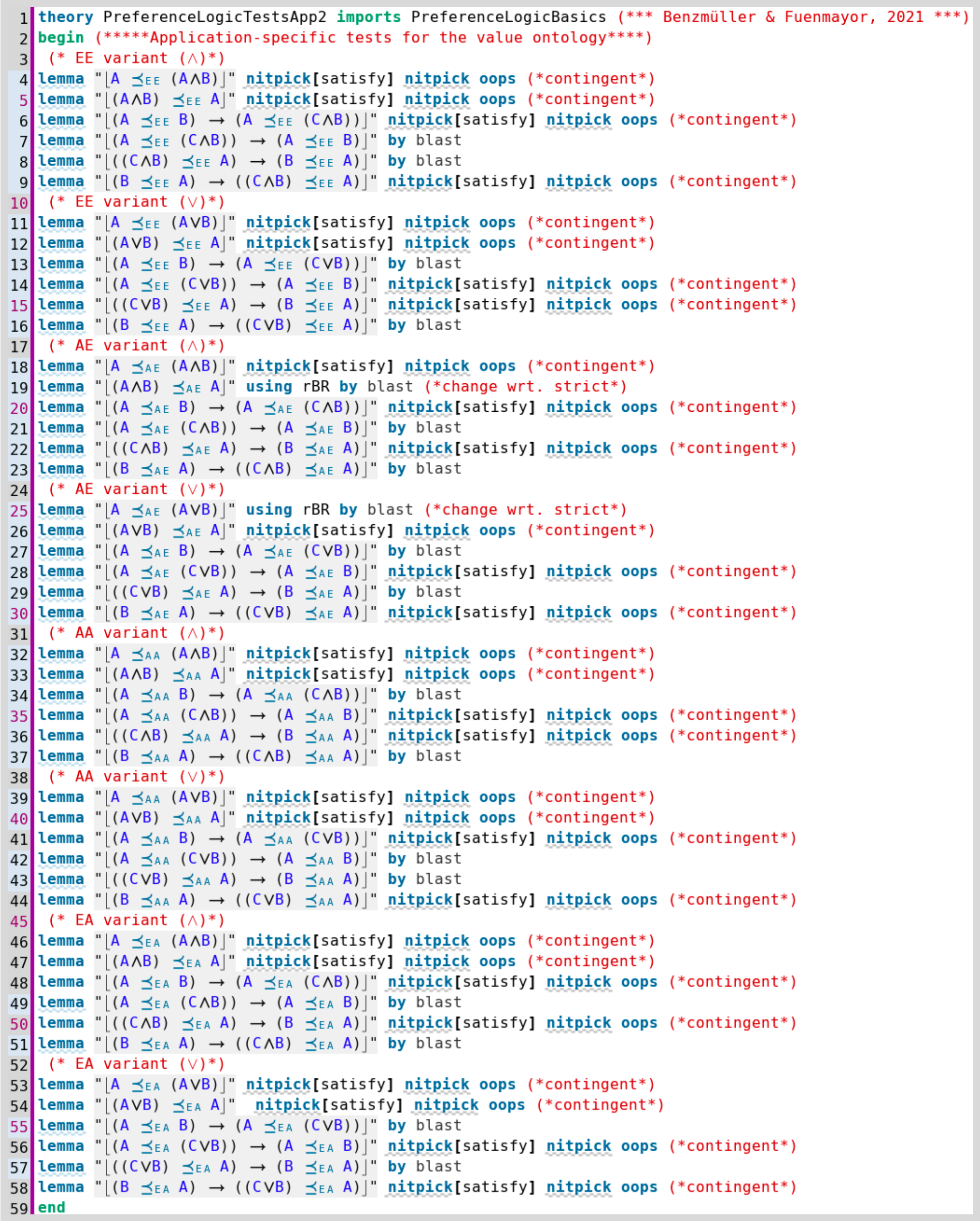
```
theory  PreferenceLogicTests2          (*** Benzmüller & Fuenmayor, 2021 ***)
   imports  PreferenceLogicCeterisParibus
begin  (*** Tests for the SSE of van Benthem et al, JPL 2009 ***)
(*** Section 5: Equality-based Ceteris Paribus Preference Logic ***)
 (*Some tests: dualities*)
lemma  "⌊(⟨Γ⟩^≼φ)  ↔  ¬([Γ]^≼¬φ)⌋"   by auto
lemma  "⌊(⟨Γ⟩^≺φ)  ↔  ¬([Γ]^≺¬φ)⌋"   by auto
lemma  "⌊(⟨Γ⟩φ)   ↔  ¬([Γ]¬φ)⌋"     by auto
 (*Lemma 2*)
lemma  lemma2_1 : "(◇^≼φ) w ⟷ ( ⟨∅⟩^≼φ) w"  by auto
lemma  lemma2_2 : "(◇^≺φ) w ⟷ ( ⟨∅⟩^≺φ) w"  by auto
lemma  lemma2_3 : "((Eφ) w ⟷ ( ⟨∅⟩φ) w) ∧ (( Aφ) w ⟷ ([ ∅]φ) w)"  by auto
(**Axiomatization:**)
 (*inclusion and interaction axioms *)
lemma  Inc1: "⌊(⟨Γ⟩^≺φ)  →  (⟨Γ⟩^≼φ)⌋" using SBR_def  by auto
lemma  Inc2: "⌊(⟨Γ⟩^≼φ)  →  (⟨Γ⟩φ)⌋"   by auto
lemma  Int3: "⌊(⟨Γ⟩^≼(⟨Γ⟩^≼φ))  →(⟨Γ⟩^≼φ)⌋"   by (meson tBR )
lemma  Int4: "⌊(⟨Γ⟩^≺(⟨Γ⟩^≼φ))  →(⟨Γ⟩^≺φ)⌋"   by (metis SBR_def tBR  )
lemma  Int5: "⌊(ψ∧(⟨Γ⟩^≼φ))  →  (((⟨Γ⟩^≺φ)∨(⟨Γ⟩^≼(φ∧(⟨Γ⟩^≼φ))))⌋" by (metis rBR )
 (*ceteris paribus reflexivity*)
lemma  CetPar6 : "φ  ∈ Γ  ⟶  ⌊(⟨Γ⟩φ)  →  φ⌋"      by blast
lemma  CetPar7 : "φ  ∈ Γ  ⟶  ⌊(⟨Γ⟩¬φ)  →  ¬φ⌋"  by blast
 (*monotonicity*)
lemma  CetPar8 :  "Γ  ⊆ Γ'  ⟶  ⌊(⟨Γ'⟩φ)  →  (⟨Γ⟩φ)⌋"     by auto
lemma  CetPar9 :  "Γ  ⊆ Γ'  ⟶  ⌊(⟨Γ'⟩^≼φ)  →  (⟨Γ⟩^≼φ)⌋"  by auto
lemma  CetPar10 : "Γ  ⊆ Γ'  ⟶  ⌊(⟨Γ'⟩^≺φ)  →  (⟨Γ⟩^≺φ)⌋"  by auto
 (*increase (decrease) of ceteris paribus sets*)
lemma  CetPar11a : "⌊(φ  ∧  (⟨Γ⟩(α  ∧  φ) ))  →  (⟨Γ∪{φ}⟩α)⌋"        by auto
lemma  CetPar11b : "⌊((¬φ)  ∧  (⟨Γ⟩(α  ∧  ¬φ)))  →  (⟨Γ∪{φ}⟩α)⌋"    by auto
lemma  CetPar12a : "⌊(φ  ∧  (⟨Γ⟩^≼(α  ∧  φ) ))  →  (⟨Γ∪{φ}⟩^≼α)⌋"      by auto
lemma  CetPar12b : "⌊((¬φ)  ∧  (⟨Γ⟩^≼(α  ∧  ¬φ)))  →  (⟨Γ∪{φ}⟩^≼α)⌋"  by auto
lemma  CetPar13a : "⌊(φ  ∧  (⟨Γ⟩^≺(α  ∧  φ) ))  →  (⟨Γ∪{φ}⟩^≺α)⌋"      by auto
lemma  CetPar13b : "⌊((¬φ)  ∧  (⟨Γ⟩^≺(α  ∧  ¬φ)))  →  (⟨Γ∪{φ}⟩^≺α)⌋"  by auto
 (*Example 1, Lemma 4, Corollary 1 and Lemma5*)
lemma  Ex1:   "⌊(([Γ]^≼φ)  ∧  (⟨Γ⟩^≼α))  →  (⟨Γ∪{φ}⟩^≼α)⌋"  using rBR  by auto
lemma  Lemma4 : "(⟨Γ⟩^≼φ) w  ⟶  (∃v. (w  ⊴_Γ v)  ∧  (φ v))"   by simp
lemma  Cor1:    "(⟨Γ⟩φ)  w  ⟶  (∃v. (w  ≡_Γ v)  ∧  (φ v))"   by simp
lemma  Lemma5 : "(w  ⊴_Γ v)  ⟷  ((w ≼ v)  ∧  (w  ≡_Γ v))"      by auto
(**** Section 6: Ceteris Paribus Counterparts ****)
 (*AA-variant (drawing upon von Wright's)*)
lemma  "(φ  ≺AA^Γ  ψ) u ⟷  (φ ≺ AA^Γ  ψ) u"  nitpick oops  (*Ctm*)
lemma  "(φ  ≺AA^Γ  ψ) u  ⟶  (φ ≺ AA^Γ  ψ) u"  nitpick oops  (*Ctm*)
lemma  "(φ ≺ AA^Γ  ψ) u  ⟶  (φ  ≺AA^Γ  ψ) u"  using SBR_def  by auto
lemma  "is_total SBR ⟶  (φ   ≺AA^Γ  ψ) u ⟷  (φ ≺ AA^Γ  ψ) u"  by (smt SBR_def )
lemma  "(φ  ≼AA^Γ  ψ) u   ⟷  (φ ≼ AA^Γ  ψ) u"  nitpick oops  (*Ctm*)
lemma  "(φ  ≼AA^Γ  ψ) u  ⟶  (φ ≼ AA^Γ  ψ) u"   nitpick oops  (*Ctm*)
lemma  "(φ ≼ AA^Γ  ψ) u  ⟶  (φ  ≼AA^Γ  ψ) u"   using SBR_def  by auto
lemma  "is_total SBR ⟶  (φ   ≼AA^Γ  ψ) u ⟷  (φ ≼ AA^Γ  ψ) u"  by (smt SBR_def )
 (*AE-variant*)
lemma  leAE_cp_pref : "(φ  ≼AE^Γ  ψ) u ⟷  (φ ≼ AE^Γ  ψ) u"  by auto
lemma  leqAE_cp_pref : "(φ  ≺AE^Γ  ψ) u ⟷  (φ ≺ AE^Γ  ψ) u"  by auto
 (*miscelaneous tests*)
lemma  "let Γ=∅ in ⌊(φ≺AA^Γ  ψ)  ↔  (φ≺AA ψ)⌋"  by simp
lemma  "let Γ={⊥} in ⌊(φ≺AA^Γ  ψ)  ↔  (φ≺AA ψ)⌋"  by simp
lemma  "let Γ={⊥,A} in ⌊(φ≺AA^Γ  ψ)  ↔  (φ≺AA ψ)⌋"  nitpick oops  (*Ctm*)
lemma  "let Γ={A} in ⌊(φ≺AA^Γ  ψ)  →  (A  →  (φ≺AA ψ))⌋"  nitpick oops  (*Ctm*)
lemma  "let Γ={A} in ⌊(A  →  (φ≺AA ψ))  →  (φ≺AA^Γ  ψ)⌋"  nitpick oops  (*Ctm*)
lemma  "let Γ=∅ in ⌊(φ≺AE^Γ  ψ)  ↔  (φ≺AE ψ)⌋"  by simp
lemma  "let Γ={⊥} in ⌊(φ≺AE^Γ  ψ)  ↔  (φ≺AE ψ)⌋"  by simp
lemma  "let Γ={⊥,A} in ⌊(φ≺AE^Γ  ψ)  ↔  (φ≺AE ψ)⌋"  nitpick oops  (*Ctm*)
lemma  "let Γ={A} in ⌊(φ≺AE^Γ  ψ)  →  (A  →  (φ≺AE ψ))⌋"  by auto
lemma  "let Γ={A,B} in ⌊(φ≺AE^Γ  ψ)  →  ((A  ∧ B)  →  (φ≺AE ψ))⌋"  by auto
lemma  "let Γ={A} in ⌊(A  →  (φ≺AE ψ))  →  (φ  ≺AE^Γ  ψ)⌋"  nitpick oops  (*Ctm*)
end
```

Figure A5. Experiments (continued): Testing the meta-theory of $\mathcal{PL}$.



```
theory PreferenceLogicTestsApp1 imports PreferenceLogicBasics (*** Benzmüller & Fuenmayor, 2021 ***)
begin (*****Application-specific tests for the value ontology****)
 (* EE variant (∧)*)
lemma "⌊A ≺EE (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≺EE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺EE B) → (A ≺EE (C∧B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺EE (C∧B)) → (A ≺EE B)⌋" by blast
lemma "⌊((C∧B) ≺EE A) → (B ≺EE A)⌋" by blast
lemma "⌊(B ≺EE A) → ((C∧B) ≺EE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
 (* EE variant (∨)*)
lemma "⌊A ≺EE (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∨B) ≺EE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺EE B) → (A ≺EE (C∨B))⌋" by blast
lemma "⌊(A ≺EE (C∨B)) → (A ≺EE B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∨B) ≺EE A) → (B ≺EE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≺EE A) → ((C∨B) ≺EE A)⌋" by blast
 (* AE variant (∧)*)
lemma "⌊A ≺AE (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≺AE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺AE B) → (A ≺AE (C∧B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺AE (C∧B)) → (A ≺AE B)⌋" by blast
lemma "⌊((C∧B) ≺AE A) → (B ≺AE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≺AE A) → ((C∧B) ≺AE A)⌋" by blast
 (* AE variant (∨)*)
lemma "⌊A ≺AE (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∨B) ≺AE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺AE B) → (A ≺AE (C∨B))⌋" by blast
lemma "⌊(A ≺AE (C∨B)) → (A ≺AE B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∨B) ≺AE A) → (B ≺AE A)⌋" by blast
lemma "⌊(B ≺AE A) → ((C∨B) ≺AE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
 (* AA variant (∧)*)
lemma "⌊A ≺AA (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≺AA A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺AA B) → (A ≺AA (C∧B))⌋" by blast
lemma "⌊(A ≺AA (C∧B)) → (A ≺AA B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∧B) ≺AA A) → (B ≺AA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≺AA A) → ((C∧B) ≺AA A)⌋" by blast
 (* AA variant (∨)*)
lemma "⌊A ≺AA (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∨B) ≺AA A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺AA B) → (A ≺AA (C∨B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺AA (C∨B)) → (A ≺AA B)⌋" by blast
lemma "⌊((C∨B) ≺AA A) → (B ≺AA A)⌋" by blast
lemma "⌊(B ≺AA A) → ((C∨B) ≺AA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
 (* EA variant (∧)*)
lemma "¬⌊A ≺EA (A∧B)⌋" using rBR by blast
lemma "⌊(A∧B) ≺EA A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺EA B) → (A ≺EA (C∧B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≺EA (C∧B)) → (A ≺EA B)⌋" by blast
lemma "⌊((C∧B) ≺EA A) → (B ≺EA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≺EA A) → ((C∧B) ≺EA A)⌋" by blast
 (* EA variant (∨)*)
lemma "⌊A ≺EA (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "¬⌊(A∨B) ≺EA A⌋" using rBR by blast
lemma "⌊(A ≺EA B) → (A ≺EA (C∨B))⌋" by blast
lemma "⌊(A ≺EA (C∨B)) → (A ≺EA B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∨B) ≺EA A) → (B ≺EA A)⌋" by blast
lemma "⌊(B ≺EA A) → ((C∨B) ≺EA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
end
```

Figure A6. Experiments (continued): Checking properties of strict preference relations.



```
theory PreferenceLogicTestsApp2 imports PreferenceLogicBasics (*** Benzmüller & Fuenmayor, 2021 ***)
begin (*****Application-specific tests for the value ontology****)
 (* EE variant (∧)*)
lemma "⌊A ≼EE (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≼EE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼EE B) → (A ≼EE (C∧B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼EE (C∧B)) → (A ≼EE B)⌋" by blast
lemma "⌊((C∧B) ≼EE A) → (B ≼EE A)⌋" by blast
lemma "⌊(B ≼EE A) → ((C∧B) ≼EE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
 (* EE variant (∨)*)
lemma "⌊A ≼EE (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∨B) ≼EE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼EE B) → (A ≼EE (C∨B))⌋" by blast
lemma "⌊(A ≼EE (C∨B)) → (A ≼EE B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∨B) ≼EE A) → (B ≼EE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≼EE A) → ((C∨B) ≼EE A)⌋" by blast
 (* AE variant (∧)*)
lemma "⌊A ≼AE (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≼AE A⌋" using rBR by blast (*change wrt. strict*)
lemma "⌊(A ≼AE B) → (A ≼AE (C∧B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼AE (C∧B)) → (A ≼AE B)⌋" by blast
lemma "⌊((C∧B) ≼AE A) → (B ≼AE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≼AE A) → ((C∧B) ≼AE A)⌋" by blast
 (* AE variant (∨)*)
lemma "⌊A ≼AE (A∨B)⌋" using rBR by blast (*change wrt. strict*)
lemma "⌊(A∨B) ≼AE A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼AE B) → (A ≼AE (C∨B))⌋" by blast
lemma "⌊(A ≼AE (C∨B)) → (A ≼AE B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∨B) ≼AE A) → (B ≼AE A)⌋" by blast
lemma "⌊(B ≼AE A) → ((C∨B) ≼AE A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
 (* AA variant (∧)*)
lemma "⌊A ≼AA (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≼AA A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼AA B) → (A ≼AA (C∧B))⌋" by blast
lemma "⌊(A ≼AA (C∧B)) → (A ≼AA B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∧B) ≼AA A) → (B ≼AA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≼AA A) → ((C∧B) ≼AA A)⌋" by blast
 (* AA variant (∨)*)
lemma "⌊A ≼AA (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∨B) ≼AA A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼AA B) → (A ≼AA (C∨B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼AA (C∨B)) → (A ≼AA B)⌋" by blast
lemma "⌊((C∨B) ≼AA A) → (B ≼AA A)⌋" by blast
lemma "⌊(B ≼AA A) → ((C∨B) ≼AA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
 (* EA variant (∧)*)
lemma "⌊A ≼EA (A∧B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∧B) ≼EA A⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼EA B) → (A ≼EA (C∧B))⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼EA (C∧B)) → (A ≼EA B)⌋" by blast
lemma "⌊((C∧B) ≼EA A) → (B ≼EA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(B ≼EA A) → ((C∧B) ≼EA A)⌋" by blast
 (* EA variant (∨)*)
lemma "⌊A ≼EA (A∨B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A∨B) ≼EA A⌋"  nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(A ≼EA B) → (A ≼EA (C∨B))⌋" by blast
lemma "⌊(A ≼EA (C∨B)) → (A ≼EA B)⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊((C∨B) ≼EA A) → (B ≼EA A)⌋" by blast
lemma "⌊(B ≼EA A) → ((C∨B) ≼EA A)⌋" nitpick[satisfy] nitpick oops (*contingent*)
end
```

Figure A7. Experiments (continued): Checking properties of strict preference relations.

Appendix A.2. Encoding of the Legal DSL (Value Ontology)

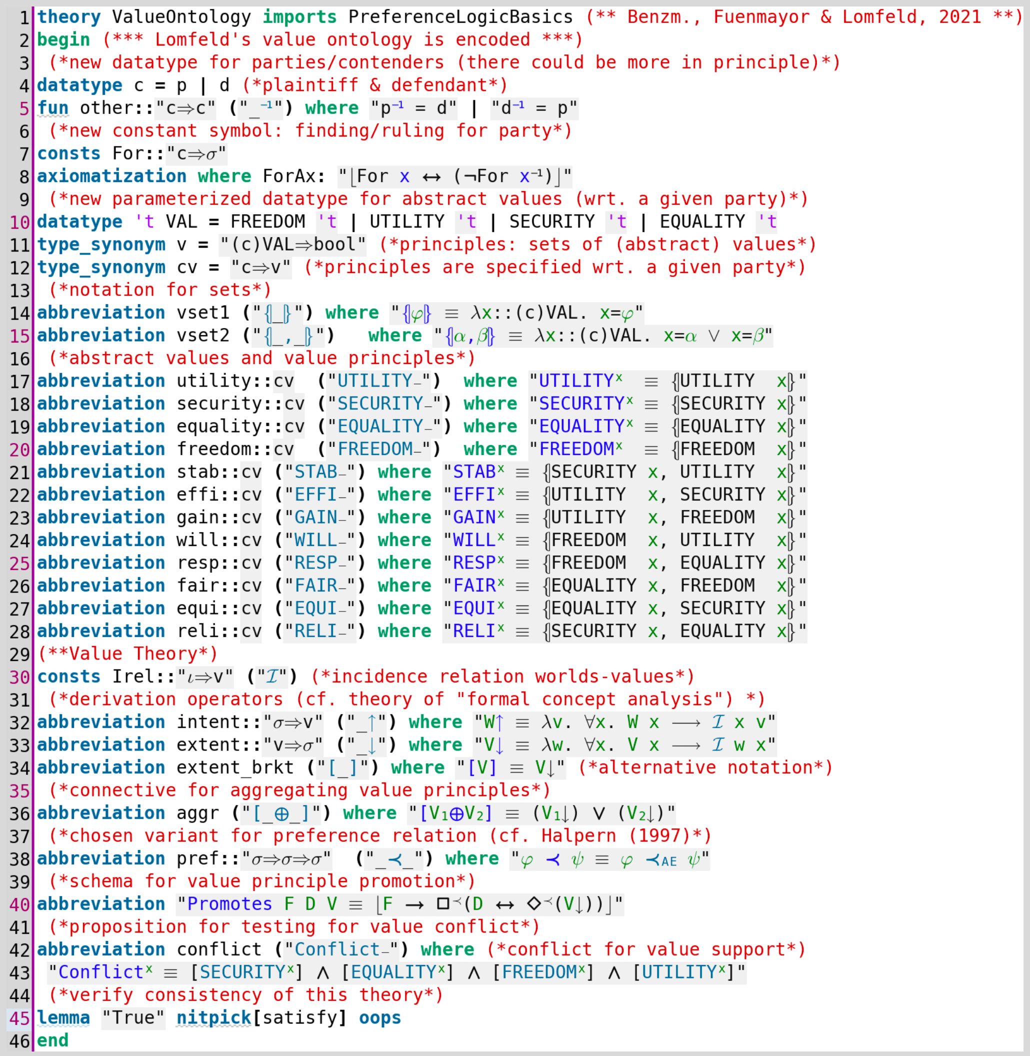
The encoding of the legal DSL (value theory or ontology) is shown in Figure A8. The new theory is termed "ValueOntology", and it imports theory "PreferenceLogicBasics" (and thus recursively also *Isabelle/HOL*'s internal theory "Main").

As a preliminary, the legal parties *plaintiff* and *defendant* are introduced as an (extensible) two-valued datatype together with a function to obtain for a given party the *other* one (x^{-1}) (Lines 4–5), and a predicate modelling the ruling *for* a party is also provided (Lines 7–8).

As regards the *discoursive grammar* value theory, a four-valued (parameterised) datatype is introduced (Line 10) as described in Section 2. Moreover, type aliases (Lines 11–12) and set-constructor operators for values (Lines 14–15) are introduced for ease of presentation. The legal principles from Section 2 are introduced as combinations of those basic values (Lines 17–28). As an illustration, the principle STABility is encoded as a set composed of the basic values SECURITY and UTILITY.

Next, the incidence relation I and operators $\uparrow$ and $\downarrow$, borrowed and adapted from formal concept analysis (FCA), are introduced (Lines 30–34).

We then define the aggregation operator $\oplus$ as $A \oplus B := (A\downarrow \vee B\downarrow)$, i.e., we select the second candidate as discussed in Section 2. And as our preference relation of choice, we select the relation $\prec_{AE}$ (Line 38).



```
 1 theory ValueOntology imports PreferenceLogicBasics (** Benzm., Fuenmayor & Lomfeld, 2021 **)
 2 begin (*** Lomfeld's value ontology is encoded ***)
 3  (*new datatype for parties/contenders (there could be more in principle)*)
 4 datatype c = p | d (*plaintiff & defendant*)
 5 fun other::"c⇒c" ("_⁻¹") where "p⁻¹ = d" | "d⁻¹ = p"
 6  (*new constant symbol: finding/ruling for party*)
 7 consts For::"c⇒σ"
 8 axiomatization where ForAx: "⌊For x ↔ (¬For x⁻¹)⌋"
 9  (*new parameterized datatype for abstract values (wrt. a given party)*)
10 datatype 't VAL = FREEDOM 't | UTILITY 't | SECURITY 't | EQUALITY 't
11 type_synonym v = "(c)VAL⇒bool" (*principles: sets of (abstract) values*)
12 type_synonym cv = "c⇒v" (*principles are specified wrt. a given party*)
13  (*notation for sets*)
14 abbreviation vset1 ("⦃_⦄") where "⦃φ⦄ ≡ λx::(c)VAL. x=φ"
15 abbreviation vset2 ("⦃_,_⦄")   where "⦃α,β⦄ ≡ λx::(c)VAL. x=α ∨ x=β"
16  (*abstract values and value principles*)
17 abbreviation utility::cv  ("UTILITY_")  where "UTILITYˣ  ≡ ⦃UTILITY  x⦄"
18 abbreviation security::cv ("SECURITY_") where "SECURITYˣ ≡ ⦃SECURITY x⦄"
19 abbreviation equality::cv ("EQUALITY_") where "EQUALITYˣ ≡ ⦃EQUALITY x⦄"
20 abbreviation freedom::cv  ("FREEDOM_")  where "FREEDOMˣ  ≡ ⦃FREEDOM  x⦄"
21 abbreviation stab::cv ("STAB_") where "STABˣ ≡ ⦃SECURITY x, UTILITY  x⦄"
22 abbreviation effi::cv ("EFFI_") where "EFFIˣ ≡ ⦃UTILITY  x, SECURITY x⦄"
23 abbreviation gain::cv ("GAIN_") where "GAINˣ ≡ ⦃UTILITY  x, FREEDOM  x⦄"
24 abbreviation will::cv ("WILL_") where "WILLˣ ≡ ⦃FREEDOM  x, UTILITY  x⦄"
25 abbreviation resp::cv ("RESP_") where "RESPˣ ≡ ⦃FREEDOM  x, EQUALITY x⦄"
26 abbreviation fair::cv ("FAIR_") where "FAIRˣ ≡ ⦃EQUALITY x, FREEDOM  x⦄"
27 abbreviation equi::cv ("EQUI_") where "EQUIˣ ≡ ⦃EQUALITY x, SECURITY x⦄"
28 abbreviation reli::cv ("RELI_") where "RELIˣ ≡ ⦃SECURITY x, EQUALITY x⦄"
29 (**Value Theory*)
30 consts Irel::"ι⇒v" ("ℐ") (*incidence relation worlds-values*)
31  (*derivation operators (cf. theory of "formal concept analysis") *)
32 abbreviation intent::"σ⇒v" ("_↑") where "W↑ ≡ λv. ∀x. W x ⟶ ℐ x v"
33 abbreviation extent::"v⇒σ" ("_↓") where "V↓ ≡ λw. ∀x. V x ⟶ ℐ w x"
34 abbreviation extent_brkt ("[_]") where "[V] ≡ V↓" (*alternative notation*)
35  (*connective for aggregating value principles*)
36 abbreviation aggr ("[_⊕_]") where "[V₁⊕V₂] ≡ (V₁↓) ∨ (V₂↓)"
37  (*chosen variant for preference relation (cf. Halpern (1997)*)
38 abbreviation pref::"σ⇒σ⇒σ"  ("_≺_") where "φ ≺ ψ ≡ φ ≺AE ψ"
39  (*schema for value principle promotion*)
40 abbreviation "Promotes F D V ≡ ⌊F → □≺(D ↔ ◇≺(V↓))⌋"
41  (*proposition for testing for value conflict*)
42 abbreviation conflict ("Conflict_") where (*conflict for value support*)
43  "Conflictˣ ≡ [SECURITYˣ] ∧ [EQUALITYˣ] ∧ [FREEDOMˣ] ∧ [UTILITYˣ]"
44  (*verify consistency of this theory*)
45 lemma "True" nitpick[satisfy] oops
46 end
```

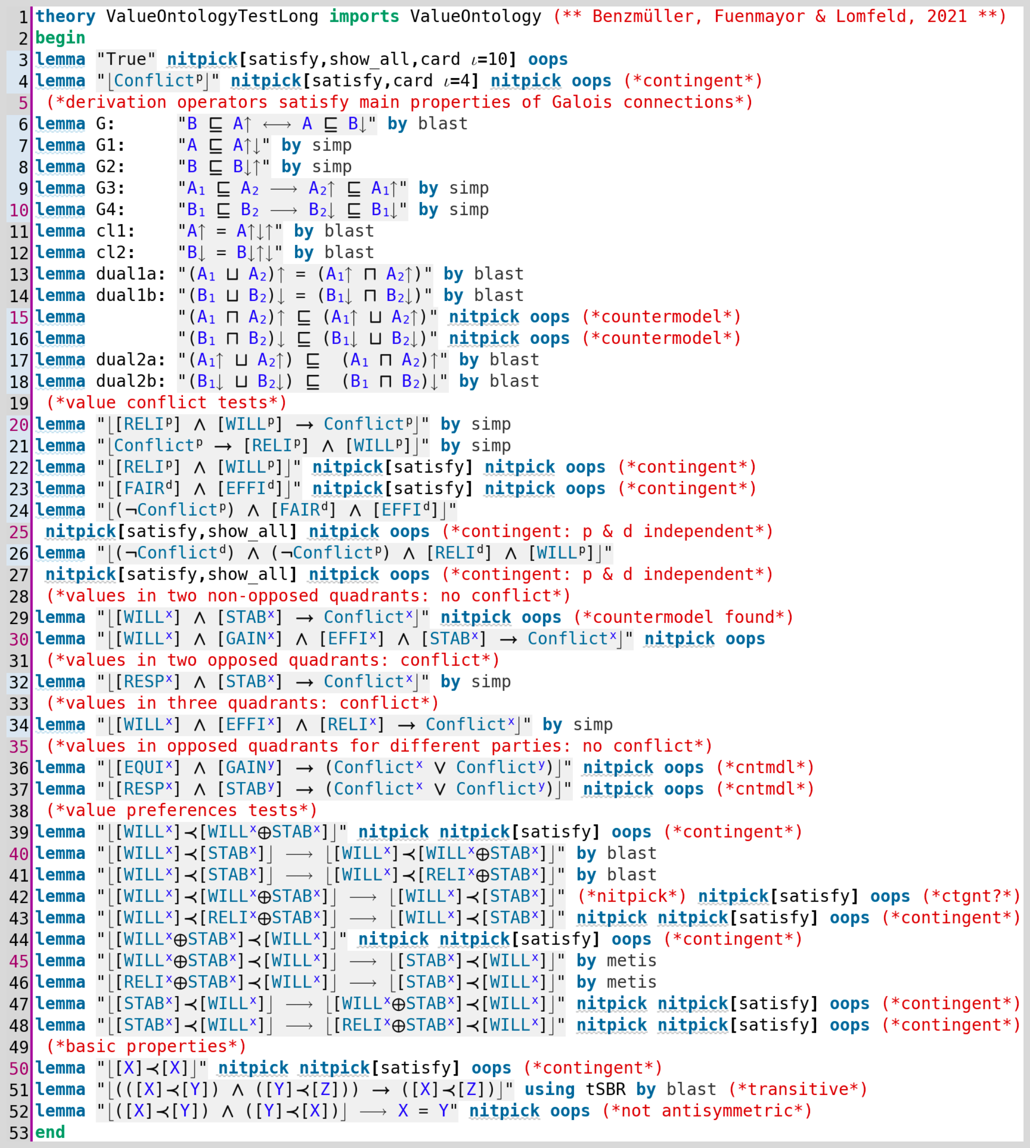
Figure A8. Encoding of the legal DSL (value ontology).

Finally, we introduce the "Promotes" schema for encoding the promotion of value principles via legal decisions (Line 40), and we introduce a notion "Conflict^x" expressing a legal value conflict for a party x (Lines 42–43).

The consistency of the theory is confirmed by *Nitpick* (Line 45).

Tests on the modelling and encoding of the legal DSL are displayed in Figure A9.

Among others, we verify that the pair of operators for *extension* ($\downarrow$) and *intension* ($\uparrow$), cf. *Formal Concept Analysis* (Ganter and Wille [93]), indeed constitute a Galois connection (Lines 6–18), and we carry out some further tests on the value theory (extending the ones presented in Figure 6) concerning value aggregation and consistency (Lines 20ff.).



```
theory ValueOntologyTestLong imports ValueOntology (** Benzmüller, Fuenmayor & Lomfeld, 2021 **)
begin
lemma "True" nitpick[satisfy,show_all,card ι=10] oops
lemma "⌊Conflictᵖ⌋" nitpick[satisfy,card ι=4] nitpick oops (*contingent*)
 (*derivation operators satisfy main properties of Galois connections*)
lemma G:      "B ⊑ A↑ ⟷ A ⊑ B↓" by blast
lemma G1:     "A ⊑ A↑↓" by simp
lemma G2:     "B ⊑ B↓↑" by simp
lemma G3:     "A₁ ⊑ A₂ ⟶ A₂↑ ⊑ A₁↑" by simp
lemma G4:     "B₁ ⊑ B₂ ⟶ B₂↓ ⊑ B₁↓" by simp
lemma cl1:    "A↑ = A↑↓↑" by blast
lemma cl2:    "B↓ = B↓↑↓" by blast
lemma dual1a: "(A₁ ⊔ A₂)↑ = (A₁↑ ⊓ A₂↑)" by blast
lemma dual1b: "(B₁ ⊔ B₂)↓ = (B₁↓ ⊓ B₂↓)" by blast
lemma         "(A₁ ⊓ A₂)↑ ⊑ (A₁↑ ⊔ A₂↑)" nitpick oops (*countermodel*)
lemma         "(B₁ ⊓ B₂)↓ ⊑ (B₁↓ ⊔ B₂↓)" nitpick oops (*countermodel*)
lemma dual2a: "(A₁↑ ⊔ A₂↑) ⊑  (A₁ ⊓ A₂)↑" by blast
lemma dual2b: "(B₁↓ ⊔ B₂↓) ⊑  (B₁ ⊓ B₂)↓" by blast
 (*value conflict tests*)
lemma "⌊[RELIᵖ] ∧ [WILLᵖ] → Conflictᵖ⌋" by simp
lemma "⌊Conflictᵖ → [RELIᵖ] ∧ [WILLᵖ]⌋" by simp
lemma "⌊[RELIᵖ] ∧ [WILLᵖ]⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊[FAIRᵈ] ∧ [EFFIᵈ]⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(¬Conflictᵖ) ∧ [FAIRᵈ] ∧ [EFFIᵈ]⌋"
 nitpick[satisfy,show_all] nitpick oops (*contingent: p & d independent*)
lemma "⌊(¬Conflictᵈ) ∧ (¬Conflictᵖ) ∧ [RELIᵈ] ∧ [WILLᵖ]⌋"
 nitpick[satisfy,show_all] nitpick oops (*contingent: p & d independent*)
 (*values in two non-opposed quadrants: no conflict*)
lemma "⌊[WILLˣ] ∧ [STABˣ] → Conflictˣ⌋" nitpick oops (*countermodel found*)
lemma "⌊[WILLˣ] ∧ [GAINˣ] ∧ [EFFIˣ] ∧ [STABˣ] → Conflictˣ⌋" nitpick oops
 (*values in two opposed quadrants: conflict*)
lemma "⌊[RESPˣ] ∧ [STABˣ] → Conflictˣ⌋" by simp
 (*values in three quadrants: conflict*)
lemma "⌊[WILLˣ] ∧ [EFFIˣ] ∧ [RELIˣ] → Conflictˣ⌋" by simp
 (*values in opposed quadrants for different parties: no conflict*)
lemma "⌊[EQUIˣ] ∧ [GAINʸ] → (Conflictˣ ∨ Conflictʸ)⌋" nitpick oops (*cntmdl*)
lemma "⌊[RESPˣ] ∧ [STABʸ] → (Conflictˣ ∨ Conflictʸ)⌋" nitpick oops (*cntmdl*)
 (*value preferences tests*)
lemma "⌊[WILLˣ]≺[WILLˣ⊕STABˣ]⌋" nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊[WILLˣ]≺[STABˣ]⌋ ⟶ ⌊[WILLˣ]≺[WILLˣ⊕STABˣ]⌋" by blast
lemma "⌊[WILLˣ]≺[STABˣ]⌋ ⟶ ⌊[WILLˣ]≺[RELIˣ⊕STABˣ]⌋" by blast
lemma "⌊[WILLˣ]≺[WILLˣ⊕STABˣ]⌋ ⟶ ⌊[WILLˣ]≺[STABˣ]⌋" (*nitpick*) nitpick[satisfy] oops (*ctgnt?*)
lemma "⌊[WILLˣ]≺[RELIˣ⊕STABˣ]⌋ ⟶ ⌊[WILLˣ]≺[STABˣ]⌋" nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊[WILLˣ⊕STABˣ]≺[WILLˣ]⌋" nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊[WILLˣ⊕STABˣ]≺[WILLˣ]⌋ ⟶ ⌊[STABˣ]≺[WILLˣ]⌋" by metis
lemma "⌊[RELIˣ⊕STABˣ]≺[WILLˣ]⌋ ⟶ ⌊[STABˣ]≺[WILLˣ]⌋" by metis
lemma "⌊[STABˣ]≺[WILLˣ]⌋ ⟶ ⌊[WILLˣ⊕STABˣ]≺[WILLˣ]⌋" nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊[STABˣ]≺[WILLˣ]⌋ ⟶ ⌊[RELIˣ⊕STABˣ]≺[WILLˣ]⌋" nitpick nitpick[satisfy] oops (*contingent*)
 (*basic properties*)
lemma "⌊[X]≺[X]⌋" nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊(([X]≺[Y]) ∧ ([Y]≺[Z])) → ([X]≺[Z])⌋" using tSBR by blast (*transitive*)
lemma "⌊([X]≺[Y]) ∧ ([Y]≺[X])⌋ ⟶ X = Y" nitpick oops (*not antisymmetric*)
end
```

Figure A9. Formally verifying/testing the legal DSL or value ontology.

Appendix A.3. Legal and World Knowledge

The encoding of the relevant legal and world knowledge (LWK) is shown in Figure A10. The defined *Isabelle/HOL* theory is termed "GeneralKowledge" and imports the "ValueOntology" (and thus recursively also "PreferenceLogicBasics") theory.

Lines 4–5 Declaration of logical constant symbols that stand for kinds of legally relevant situations.

Lines 8–11 Meaning postulates for these kinds of legally relevant situations are introduced.

Lines 14–16 Preference relations for these kinds of legally relevant situations are introduced.

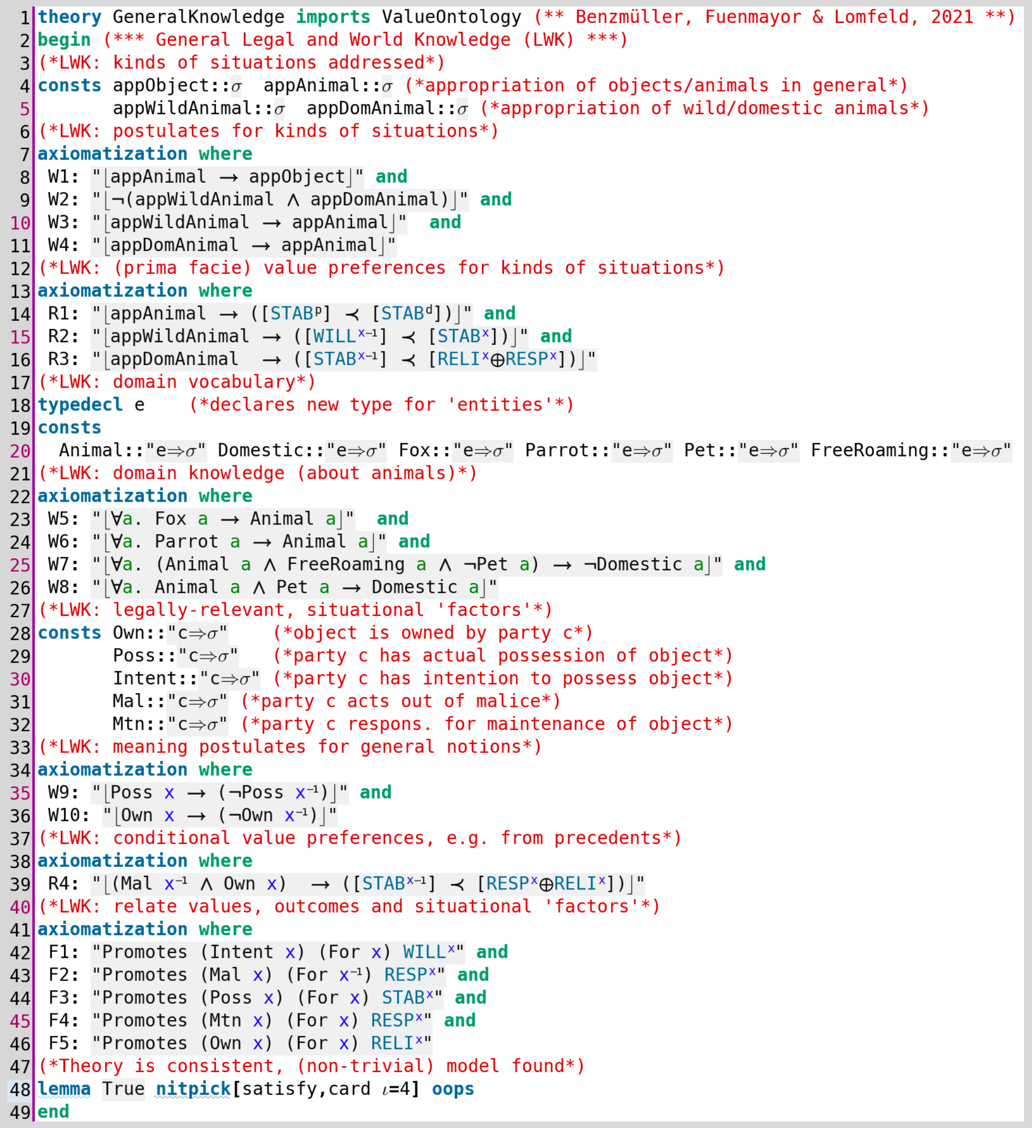
Lines 18–26 Some simple vocabulary is introduced and some taxonomic relations for wild and domestic animals are specified.

Lines 28–36 Some relevant situational *factors* are declared and subsequently constrained by meaning postulates.

Line 39 An example for a value preference conditioned on *factors* is specified.

Lines 41–46 The situational *factors* are related with values and with ruling outcomes according to the notion of value *promotion*.

Line 48 The model finder *Nitpick* is used to confirm the consistency of the introduced theory.



```
 1 theory GeneralKnowledge imports ValueOntology (** Benzmüller, Fuenmayor & Lomfeld, 2021 **)
 2 begin (*** General Legal and World Knowledge (LWK) ***)
 3 (*LWK: kinds of situations addressed*)
 4 consts appObject::σ  appAnimal::σ (*appropriation of objects/animals in general*)
 5        appWildAnimal::σ  appDomAnimal::σ (*appropriation of wild/domestic animals*)
 6 (*LWK: postulates for kinds of situations*)
 7 axiomatization where
 8  W1: "⌊appAnimal ⟶ appObject⌋" and
 9  W2: "⌊¬(appWildAnimal ∧ appDomAnimal)⌋" and
10  W3: "⌊appWildAnimal ⟶ appAnimal⌋"  and
11  W4: "⌊appDomAnimal ⟶ appAnimal⌋"
12 (*LWK: (prima facie) value preferences for kinds of situations*)
13 axiomatization where
14  R1: "⌊appAnimal ⟶ ([STABᵖ] ≺ [STABᵈ])⌋" and
15  R2: "⌊appWildAnimal ⟶ ([WILLˣ⁻¹] ≺ [STABˣ])⌋" and
16  R3: "⌊appDomAnimal  ⟶ ([STABˣ⁻¹] ≺ [RELIˣ⊕RESPˣ])⌋"
17 (*LWK: domain vocabulary*)
18 typedecl e    (*declares new type for 'entities'*)
19 consts
20   Animal::"e⇒σ" Domestic::"e⇒σ" Fox::"e⇒σ" Parrot::"e⇒σ" Pet::"e⇒σ" FreeRoaming::"e⇒σ"
21 (*LWK: domain knowledge (about animals)*)
22 axiomatization where
23  W5: "⌊∀a. Fox a ⟶ Animal a⌋"  and
24  W6: "⌊∀a. Parrot a ⟶ Animal a⌋" and
25  W7: "⌊∀a. (Animal a ∧ FreeRoaming a ∧ ¬Pet a) ⟶ ¬Domestic a⌋" and
26  W8: "⌊∀a. Animal a ∧ Pet a ⟶ Domestic a⌋"
27 (*LWK: legally-relevant, situational 'factors'*)
28 consts Own::"c⇒σ"    (*object is owned by party c*)
29        Poss::"c⇒σ"   (*party c has actual possession of object*)
30        Intent::"c⇒σ" (*party c has intention to possess object*)
31        Mal::"c⇒σ" (*party c acts out of malice*)
32        Mtn::"c⇒σ" (*party c respons. for maintenance of object*)
33 (*LWK: meaning postulates for general notions*)
34 axiomatization where
35  W9: "⌊Poss x ⟶ (¬Poss x⁻¹)⌋" and
36  W10: "⌊Own x ⟶ (¬Own x⁻¹)⌋"
37 (*LWK: conditional value preferences, e.g. from precedents*)
38 axiomatization where
39  R4: "⌊(Mal x⁻¹ ∧ Own x)  ⟶ ([STABˣ⁻¹] ≺ [RESPˣ⊕RELIˣ])⌋"
40 (*LWK: relate values, outcomes and situational 'factors'*)
41 axiomatization where
42  F1: "Promotes (Intent x) (For x) WILLˣ" and
43  F2: "Promotes (Mal x) (For x⁻¹) RESPˣ" and
44  F3: "Promotes (Poss x) (For x) STABˣ" and
45  F4: "Promotes (Mtn x) (For x) RESPˣ" and
46  F5: "Promotes (Own x) (For x) RELIˣ"
47 (*Theory is consistent, (non-trivial) model found*)
48 lemma True nitpick[satisfy,card ι=4] oops
49 end
```

Figure A10. Encoding of relevant legal and world knowledge.

Appendix A.4. Modelling Pierson v. Post

The *Isabelle/HOL* encoding of two scenarios in the Pierson v. Post case is presented in Figures A11 and A12.

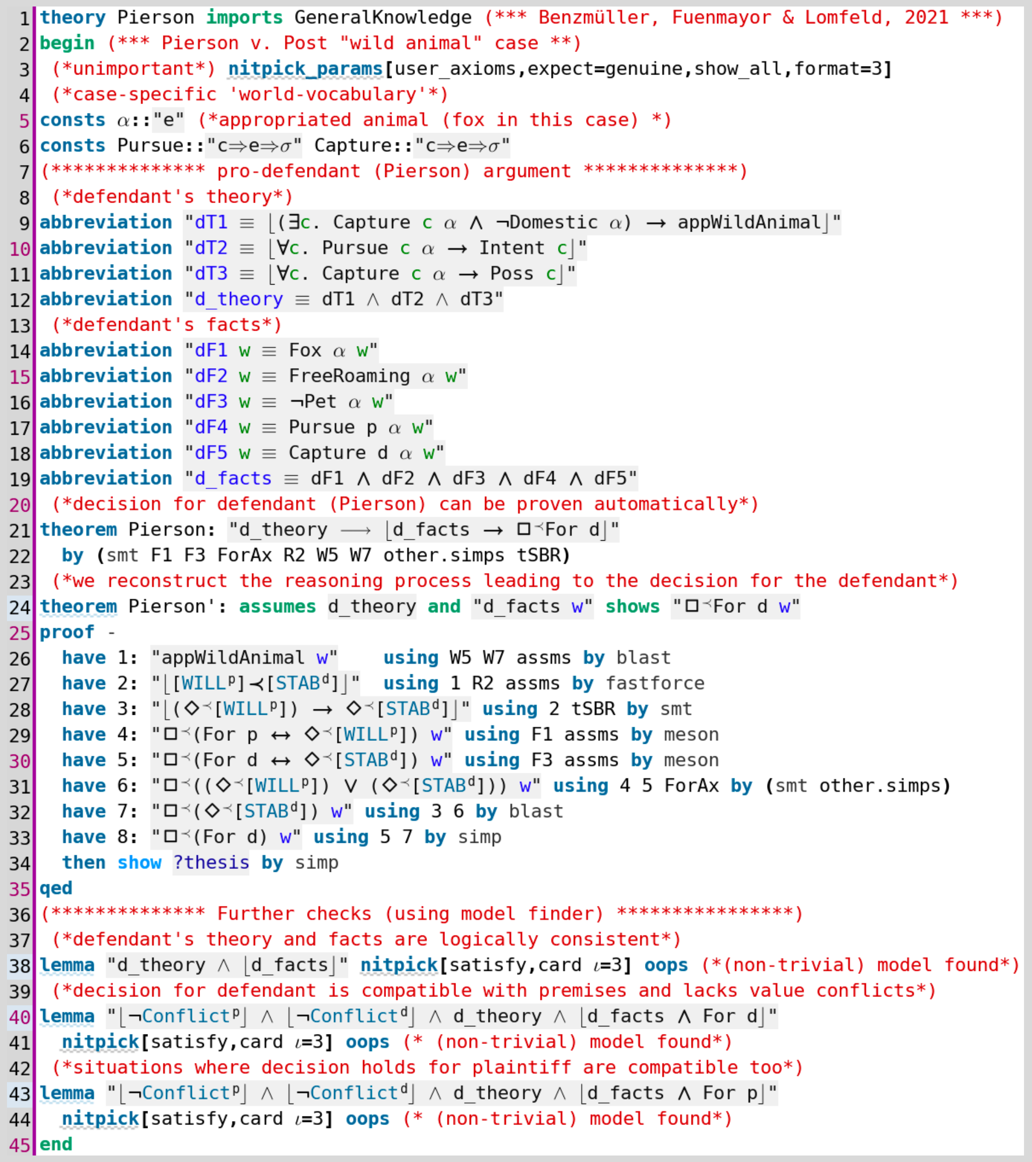
In Figure A11, which presents the initial ruling in favour of Pierson, the *Isabelle/HOL* theory is termed "Pierson" and imports the theory "GeneralKnowledge" (which recursively imports theories "ValueOntology" and "PreferenceLogicBasics").

Lines 5–19 (Generic) theory and (contingent) facts suitable to the defendant (Pierson) are postulated.

Lines 21–22 Automated proof justifying the ruling for Pierson; the dependencies of the proof are shown.

Lines 24–35 Corresponding interactive proof (with the same dependencies as for the automated one) modelling the argument justifying the finding for Pierson.

Lines 36–44 Various checks for the consistency of the assumptions and the absence of value conflicts.



```
theory Pierson imports GeneralKnowledge (*** Benzmüller, Fuenmayor & Lomfeld, 2021 ***)
begin (*** Pierson v. Post "wild animal" case **)
 (*unimportant*) nitpick_params[user_axioms,expect=genuine,show_all,format=3]
 (*case-specific 'world-vocabulary'*)
consts α::"e" (*appropriated animal (fox in this case) *)
consts Pursue::"c⇒e⇒σ" Capture::"c⇒e⇒σ"
(************* pro-defendant (Pierson) argument *************)
 (*defendant's theory*)
abbreviation "dT1 ≡ ⌊(∃c. Capture c α ∧ ¬Domestic α) → appWildAnimal⌋"
abbreviation "dT2 ≡ ⌊∀c. Pursue c α → Intent c⌋"
abbreviation "dT3 ≡ ⌊∀c. Capture c α → Poss c⌋"
abbreviation "d_theory ≡ dT1 ∧ dT2 ∧ dT3"
 (*defendant's facts*)
abbreviation "dF1 w ≡ Fox α w"
abbreviation "dF2 w ≡ FreeRoaming α w"
abbreviation "dF3 w ≡ ¬Pet α w"
abbreviation "dF4 w ≡ Pursue p α w"
abbreviation "dF5 w ≡ Capture d α w"
abbreviation "d_facts ≡ dF1 ∧ dF2 ∧ dF3 ∧ dF4 ∧ dF5"
 (*decision for defendant (Pierson) can be proven automatically*)
theorem Pierson: "d_theory ⟶ ⌊d_facts → □≺For d⌋"
  by (smt F1 F3 ForAx R2 W5 W7 other.simps tSBR)
 (*we reconstruct the reasoning process leading to the decision for the defendant*)
theorem Pierson': assumes d_theory and "d_facts w" shows "□≺For d w"
proof -
  have 1: "appWildAnimal w"    using W5 W7 assms by blast
  have 2: "⌊[WILLᵖ]≺[STABᵈ]⌋"  using 1 R2 assms by fastforce
  have 3: "⌊(◇≺[WILLᵖ]) → ◇≺[STABᵈ]⌋" using 2 tSBR by smt
  have 4: "□≺(For p ↔ ◇≺[WILLᵖ]) w" using F1 assms by meson
  have 5: "□≺(For d ↔ ◇≺[STABᵈ]) w" using F3 assms by meson
  have 6: "□≺((◇≺[WILLᵖ]) ∨ (◇≺[STABᵈ])) w" using 4 5 ForAx by (smt other.simps)
  have 7: "□≺(◇≺[STABᵈ]) w" using 3 6 by blast
  have 8: "□≺(For d) w" using 5 7 by simp
  then show ?thesis by simp
qed
(************* Further checks (using model finder) ***************)
 (*defendant's theory and facts are logically consistent*)
lemma "d_theory ∧ ⌊d_facts⌋" nitpick[satisfy,card ι=3] oops (*(non-trivial) model found*)
 (*decision for defendant is compatible with premises and lacks value conflicts*)
lemma "⌊¬Conflictᵖ⌋ ∧ ⌊¬Conflictᵈ⌋ ∧ d_theory ∧ ⌊d_facts ∧ For d⌋"
  nitpick[satisfy,card ι=3] oops (* (non-trivial) model found*)
 (*situations where decision holds for plaintiff are compatible too*)
lemma "⌊¬Conflictᵖ⌋ ∧ ⌊¬Conflictᵈ⌋ ∧ d_theory ∧ ⌊d_facts ∧ For p⌋"
  nitpick[satisfy,card ι=3] oops (* (non-trivial) model found*)
end
```

Figure A11. Modelling the Pierson vs. Post case; ruling for Pierson.

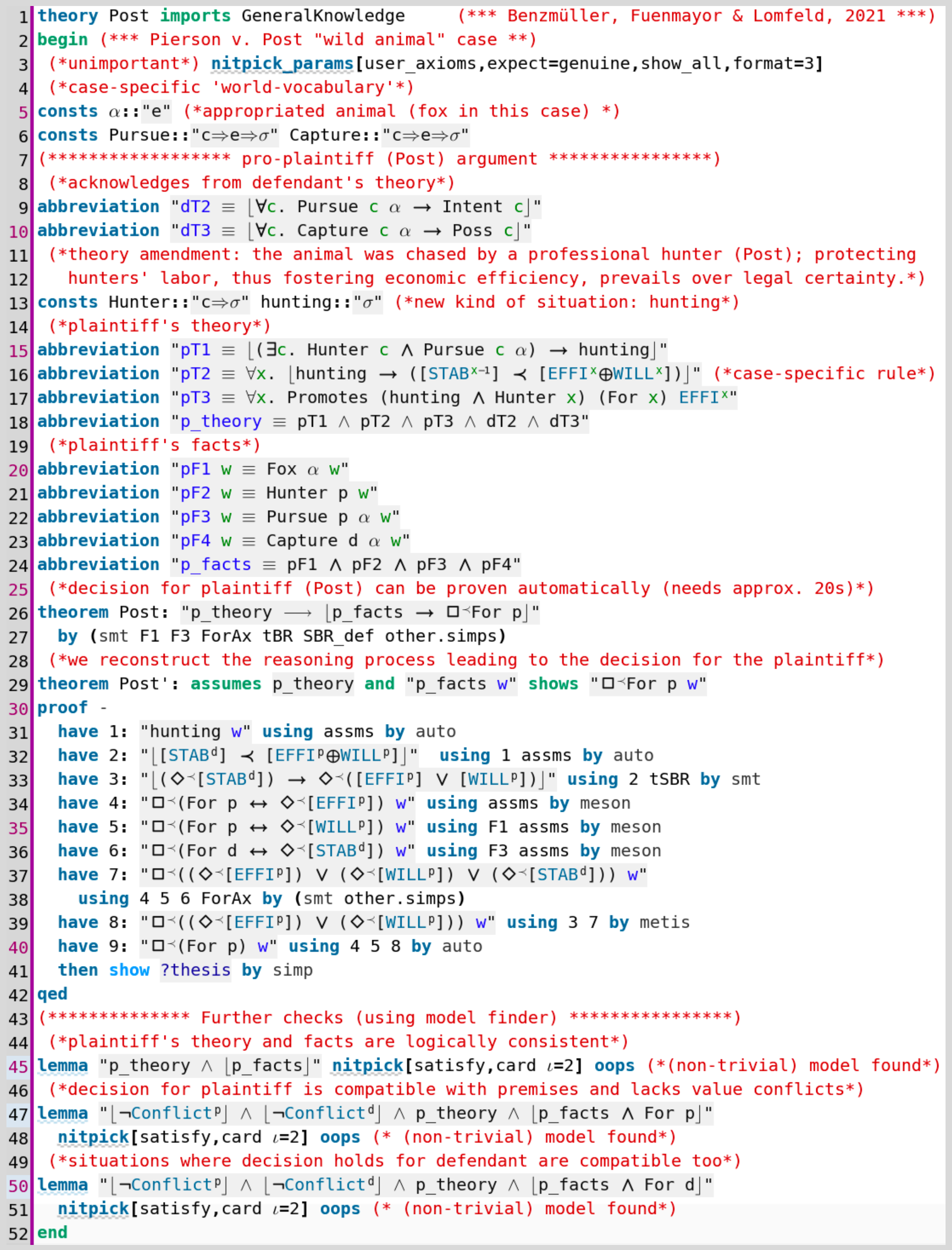
As a further illustration, we present in Figure A12 a plausible counterargument by Post. The *Isabelle/HOL* theory is now termed "Post" and imports the theory "GeneralKnowledge" (which recursively imports theories "ValueOntology" and "PreferenceLogicBasics").

Lines 5–24 Theory and facts suitable to the plaintiff (Post) are postulated.

Lines 26–27 Automated proof justifying the ruling for Post; the dependencies of the proof are shown.

Lines 29–42 Corresponding interactive proof (with the same dependencies as for the automated one) modelling the argument justifying the finding for Post.

Lines 43–51 Various checks for consistency of the assumptions and the absence of value conflicts.

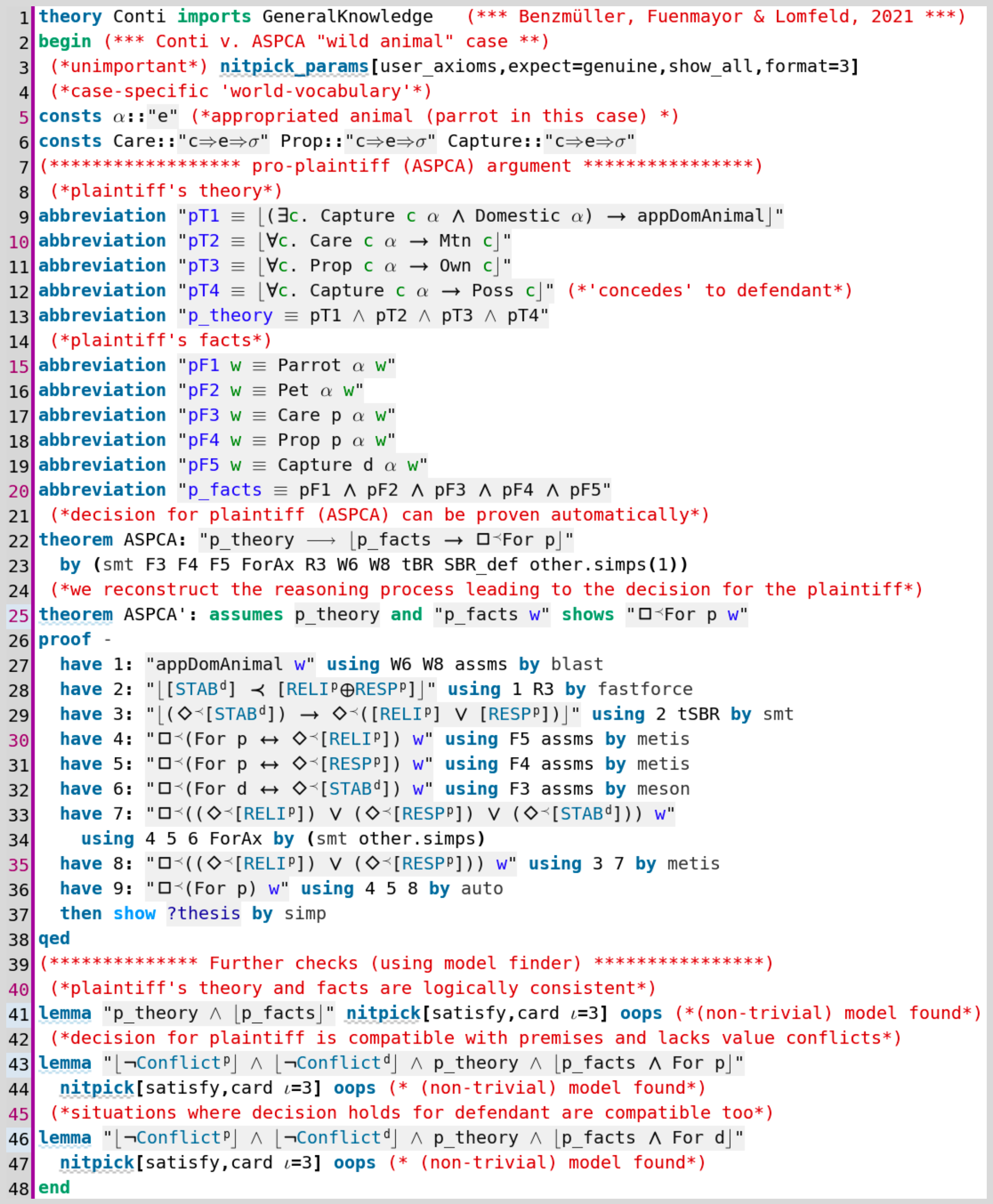


```
theory Post imports GeneralKnowledge     (*** Benzmüller, Fuenmayor & Lomfeld, 2021 ***)
begin (*** Pierson v. Post "wild animal" case **)
 (*unimportant*) nitpick_params[user_axioms,expect=genuine,show_all,format=3]
 (*case-specific 'world-vocabulary'*)
consts α::"e" (*appropriated animal (fox in this case) *)
consts Pursue::"c⇒e⇒σ" Capture::"c⇒e⇒σ"
(***************** pro-plaintiff (Post) argument ***************)
 (*acknowledges from defendant's theory*)
abbreviation "dT2 ≡ ⌊∀c. Pursue c α → Intent c⌋"
abbreviation "dT3 ≡ ⌊∀c. Capture c α → Poss c⌋"
 (*theory amendment: the animal was chased by a professional hunter (Post); protecting
   hunters' labor, thus fostering economic efficiency, prevails over legal certainty.*)
consts Hunter::"c⇒σ" hunting::"σ" (*new kind of situation: hunting*)
 (*plaintiff's theory*)
abbreviation "pT1 ≡ ⌊(∃c. Hunter c ∧ Pursue c α) → hunting⌋"
abbreviation "pT2 ≡ ∀x. ⌊hunting → ([STABˣ⁻¹] ≺ [EFFIˣ⊕WILLˣ])⌋" (*case-specific rule*)
abbreviation "pT3 ≡ ∀x. Promotes (hunting ∧ Hunter x) (For x) EFFIˣ"
abbreviation "p_theory ≡ pT1 ∧ pT2 ∧ pT3 ∧ dT2 ∧ dT3"
 (*plaintiff's facts*)
abbreviation "pF1 w ≡ Fox α w"
abbreviation "pF2 w ≡ Hunter p w"
abbreviation "pF3 w ≡ Pursue p α w"
abbreviation "pF4 w ≡ Capture d α w"
abbreviation "p_facts ≡ pF1 ∧ pF2 ∧ pF3 ∧ pF4"
 (*decision for plaintiff (Post) can be proven automatically (needs approx. 20s)*)
theorem Post: "p_theory ⟶ ⌊p_facts → □≺For p⌋"
  by (smt F1 F3 ForAx tBR SBR_def other.simps)
 (*we reconstruct the reasoning process leading to the decision for the plaintiff*)
theorem Post': assumes p_theory and "p_facts w" shows "□≺For p w"
proof -
  have 1: "hunting w" using assms by auto
  have 2: "⌊[STABᵈ] ≺ [EFFIᵖ⊕WILLᵖ]⌋"  using 1 assms by auto
  have 3: "⌊(◇≺[STABᵈ]) → ◇≺([EFFIᵖ] ∨ [WILLᵖ])⌋" using 2 tSBR by smt
  have 4: "□≺(For p ↔ ◇≺[EFFIᵖ]) w" using assms by meson
  have 5: "□≺(For p ↔ ◇≺[WILLᵖ]) w" using F1 assms by meson
  have 6: "□≺(For d ↔ ◇≺[STABᵈ]) w" using F3 assms by meson
  have 7: "□≺((◇≺[EFFIᵖ]) ∨ (◇≺[WILLᵖ]) ∨ (◇≺[STABᵈ])) w"
    using 4 5 6 ForAx by (smt other.simps)
  have 8: "□≺((◇≺[EFFIᵖ]) ∨ (◇≺[WILLᵖ])) w" using 3 7 by metis
  have 9: "□≺(For p) w" using 4 5 8 by auto
  then show ?thesis by simp
qed
(************** Further checks (using model finder) ***************)
 (*plaintiff's theory and facts are logically consistent*)
lemma "p_theory ∧ ⌊p_facts⌋" nitpick[satisfy,card ι=2] oops (*(non-trivial) model found*)
 (*decision for plaintiff is compatible with premises and lacks value conflicts*)
lemma "⌊¬Conflictᵖ⌋ ∧ ⌊¬Conflictᵈ⌋ ∧ p_theory ∧ ⌊p_facts ∧ For p⌋"
  nitpick[satisfy,card ι=2] oops (* (non-trivial) model found*)
 (*situations where decision holds for defendant are compatible too*)
lemma "⌊¬Conflictᵖ⌋ ∧ ⌊¬Conflictᵈ⌋ ∧ p_theory ∧ ⌊p_facts ∧ For d⌋"
  nitpick[satisfy,card ι=2] oops (* (non-trivial) model found*)
end
```

Figure A12. Modelling the Pierson v. Post case; ruling for Post.

Appendix A.5. Modelling Conti v. ASPCA

The reconstructed theory for the Conti v. ASPCA case is displayed in Figure A13. The *Isabelle/HOL* theory is termed "Conti" and imports the theory "GeneralKnowledge" (which recursively imports theories "ValueOntology" and "PreferenceLogicBasics").



```
 1 theory Conti imports GeneralKnowledge   (*** Benzmüller, Fuenmayor & Lomfeld, 2021 ***)
 2 begin (*** Conti v. ASPCA "wild animal" case **)
 3  (*unimportant*) nitpick_params[user_axioms,expect=genuine,show_all,format=3]
 4  (*case-specific 'world-vocabulary'*)
 5 consts α::"e" (*appropriated animal (parrot in this case) *)
 6 consts Care::"c⇒e⇒σ" Prop::"c⇒e⇒σ" Capture::"c⇒e⇒σ"
 7 (***************** pro-plaintiff (ASPCA) argument ****************)
 8  (*plaintiff's theory*)
 9 abbreviation "pT1 ≡ ⌊(∃c. Capture c α ∧ Domestic α) → appDomAnimal⌋"
10 abbreviation "pT2 ≡ ⌊∀c. Care c α → Mtn c⌋"
11 abbreviation "pT3 ≡ ⌊∀c. Prop c α → Own c⌋"
12 abbreviation "pT4 ≡ ⌊∀c. Capture c α → Poss c⌋" (*'concedes' to defendant*)
13 abbreviation "p_theory ≡ pT1 ∧ pT2 ∧ pT3 ∧ pT4"
14  (*plaintiff's facts*)
15 abbreviation "pF1 w ≡ Parrot α w"
16 abbreviation "pF2 w ≡ Pet α w"
17 abbreviation "pF3 w ≡ Care p α w"
18 abbreviation "pF4 w ≡ Prop p α w"
19 abbreviation "pF5 w ≡ Capture d α w"
20 abbreviation "p_facts ≡ pF1 ∧ pF2 ∧ pF3 ∧ pF4 ∧ pF5"
21  (*decision for plaintiff (ASPCA) can be proven automatically*)
22 theorem ASPCA: "p_theory ⟶ ⌊p_facts → □≺For p⌋"
23   by (smt F3 F4 F5 ForAx R3 W6 W8 tBR SBR_def other.simps(1))
24  (*we reconstruct the reasoning process leading to the decision for the plaintiff*)
25 theorem ASPCA': assumes p_theory and "p_facts w" shows "□≺For p w"
26 proof -
27   have 1: "appDomAnimal w" using W6 W8 assms by blast
28   have 2: "⌊[STAB^d] ≺ [RELI^p⊕RESP^p]⌋" using 1 R3 by fastforce
29   have 3: "⌊(◇≺[STAB^d]) → ◇≺([RELI^p] ∨ [RESP^p])⌋" using 2 tSBR by smt
30   have 4: "□≺(For p ↔ ◇≺[RELI^p]) w" using F5 assms by metis
31   have 5: "□≺(For p ↔ ◇≺[RESP^p]) w" using F4 assms by metis
32   have 6: "□≺(For d ↔ ◇≺[STAB^d]) w" using F3 assms by meson
33   have 7: "□≺((◇≺[RELI^p]) ∨ (◇≺[RESP^p]) ∨ (◇≺[STAB^d])) w"
34     using 4 5 6 ForAx by (smt other.simps)
35   have 8: "□≺((◇≺[RELI^p]) ∨ (◇≺[RESP^p])) w" using 3 7 by metis
36   have 9: "□≺(For p) w" using 4 5 8 by auto
37   then show ?thesis by simp
38 qed
39 (************** Further checks (using model finder) ****************)
40  (*plaintiff's theory and facts are logically consistent*)
41 lemma "p_theory ∧ ⌊p_facts⌋" nitpick[satisfy,card ι=3] oops (*(non-trivial) model found*)
42  (*decision for plaintiff is compatible with premises and lacks value conflicts*)
43 lemma "⌊¬Conflict^p⌋ ∧ ⌊¬Conflict^d⌋ ∧ p_theory ∧ ⌊p_facts ∧ For p⌋"
44   nitpick[satisfy,card ι=3] oops (* (non-trivial) model found*)
45  (*situations where decision holds for defendant are compatible too*)
46 lemma "⌊¬Conflict^p⌋ ∧ ⌊¬Conflict^d⌋ ∧ p_theory ∧ ⌊p_facts ∧ For d⌋"
47   nitpick[satisfy,card ι=3] oops (* (non-trivial) model found*)
48 end
```

Figure A13. Modelling of the Conti v. ASPCA case.

Lines 5–20 The theory and the facts of the pro-plaintiff (ASPCA) argument are formulated.

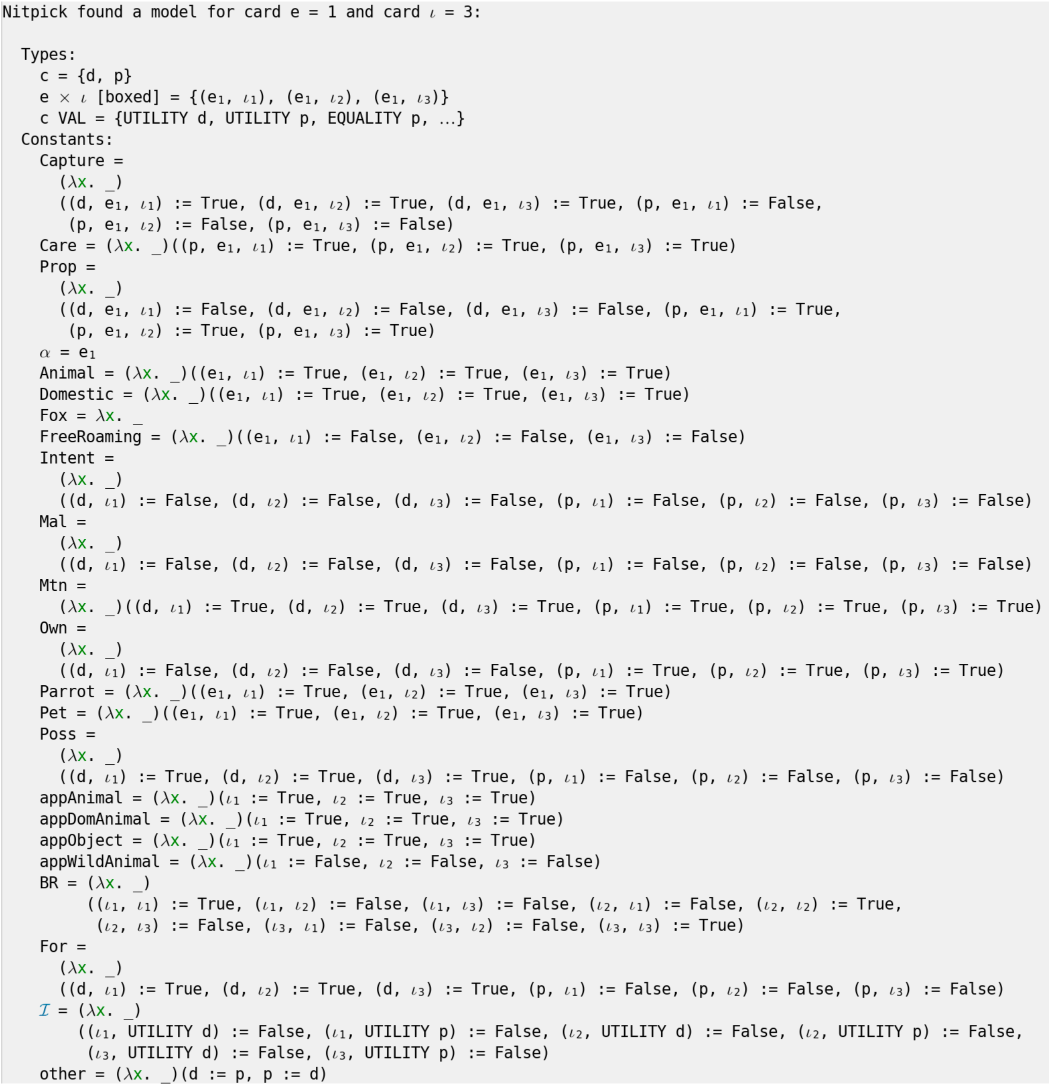
Lines 22–23 Automated proof justifying the ruling for ASPCA; the dependencies of the proof are shown.

Lines 25–38 Corresponding interactive proof (with the same dependencies as for the automated one) modelling the argument justifying the finding for ASPCA.

Lines 39–47 Various checks for consistency of the assumptions and the absence of value conflicts.

Appendix A.6. Complex (Counter-)Models

In Figure A14, we present an example of a model computed by model finder *Nitpick* for the statement in Line 41 in Figure A13. This non-trivial model features three possible worlds/states. It illustrates the richness of the information and the level of detail that is supported in the model (and countermodel) finding technology for HOL. This information is very helpful to support the knowledge engineer and user of the LogiKEy framework to gain insight about the modelled structures. We observe that the proof assistant *Isabelle/HOL* allows for the parallel execution of its integrated tools. We can thus execute, for a given candidate theorem, all three tasks in parallel (and in different modes): theorem proving, model finding, and countermodel finding. This is one reason for the very good response rates we have experienced in our work with the system—despite the general undecidability of HOL.



```
Nitpick found a model for card e = 1 and card ι = 3:

  Types:
    c = {d, p}
    e × ι [boxed] = {(e₁, ι₁), (e₁, ι₂), (e₁, ι₃)}
    c VAL = {UTILITY d, UTILITY p, EQUALITY p, …}
  Constants:
    Capture =
      (λx. _)
      ((d, e₁, ι₁) := True, (d, e₁, ι₂) := True, (d, e₁, ι₃) := True, (p, e₁, ι₁) := False,
       (p, e₁, ι₂) := False, (p, e₁, ι₃) := False)
    Care = (λx. _)((p, e₁, ι₁) := True, (p, e₁, ι₂) := True, (p, e₁, ι₃) := True)
    Prop =
      (λx. _)
      ((d, e₁, ι₁) := False, (d, e₁, ι₂) := False, (d, e₁, ι₃) := False, (p, e₁, ι₁) := True,
       (p, e₁, ι₂) := True, (p, e₁, ι₃) := True)
    α = e₁
    Animal = (λx. _)((e₁, ι₁) := True, (e₁, ι₂) := True, (e₁, ι₃) := True)
    Domestic = (λx. _)((e₁, ι₁) := True, (e₁, ι₂) := True, (e₁, ι₃) := True)
    Fox = λx. _
    FreeRoaming = (λx. _)((e₁, ι₁) := False, (e₁, ι₂) := False, (e₁, ι₃) := False)
    Intent =
      (λx. _)
      ((d, ι₁) := False, (d, ι₂) := False, (d, ι₃) := False, (p, ι₁) := False, (p, ι₂) := False, (p, ι₃) := False)
    Mal =
      (λx. _)
      ((d, ι₁) := False, (d, ι₂) := False, (d, ι₃) := False, (p, ι₁) := False, (p, ι₂) := False, (p, ι₃) := False)
    Mtn =
      (λx. _)((d, ι₁) := True, (d, ι₂) := True, (d, ι₃) := True, (p, ι₁) := True, (p, ι₂) := True, (p, ι₃) := True)
    Own =
      (λx. _)
      ((d, ι₁) := False, (d, ι₂) := False, (d, ι₃) := False, (p, ι₁) := True, (p, ι₂) := True, (p, ι₃) := True)
    Parrot = (λx. _)((e₁, ι₁) := True, (e₁, ι₂) := True, (e₁, ι₃) := True)
    Pet = (λx. _)((e₁, ι₁) := True, (e₁, ι₂) := True, (e₁, ι₃) := True)
    Poss =
      (λx. _)
      ((d, ι₁) := True, (d, ι₂) := True, (d, ι₃) := True, (p, ι₁) := False, (p, ι₂) := False, (p, ι₃) := False)
    appAnimal = (λx. _)(ι₁ := True, ι₂ := True, ι₃ := True)
    appDomAnimal = (λx. _)(ι₁ := True, ι₂ := True, ι₃ := True)
    appObject = (λx. _)(ι₁ := True, ι₂ := True, ι₃ := True)
    appWildAnimal = (λx. _)(ι₁ := False, ι₂ := False, ι₃ := False)
    BR = (λx. _)
         ((ι₁, ι₁) := True, (ι₁, ι₂) := False, (ι₁, ι₃) := False, (ι₂, ι₁) := False, (ι₂, ι₂) := True,
          (ι₂, ι₃) := False, (ι₃, ι₁) := False, (ι₃, ι₂) := False, (ι₃, ι₃) := True)
    For =
      (λx. _)
      ((d, ι₁) := True, (d, ι₂) := True, (d, ι₃) := True, (p, ι₁) := False, (p, ι₂) := False, (p, ι₃) := False)
    𝓘 = (λx. _)
        ((ι₁, UTILITY d) := False, (ι₁, UTILITY p) := False, (ι₂, UTILITY d) := False, (ι₂, UTILITY p) := False,
         (ι₃, UTILITY d) := False, (ι₃, UTILITY p) := False)
    other = (λx. _)(d := p, p := d)
```

Figure A14. Example of a (satisfying) model to the statement in Line 26 in Figure A13.

Notes

[1] In Section 6, these values will be assigned to particular parties/actors so that ruling in favour of different parties may promote different values.

[2] All these taxonomies are pluralist frameworks that do encompass differences in global value patterns and cultural value evolution (Hofstede [47], Inglehart [48]). For an approach oriented at Maslow's hierarchy of needs (Bench-Capon [49]).

[3] Shallow semantical embeddings are different from *deep embeddings* of an object logic. In the latter case the syntax of the object logic is represented using an inductive data structure (e.g., following the definition of the language). The semantics of a formula

is then evaluated by recursively traversing the data structure, and additionally a proof theory for the logic maybe be encoded. Deep embeddings typically require technical inductive proofs, which hinder proof automation, that can be avoided when shallow semantical embeddings are used instead. For more information on shallow and deep embeddings we refer to the literature (Gibbons and Wu [59], Svenningsson and Axelsson [60]).

[4] In some cases, it can be convenient to split one or more layers into sublayers. For instance, in our case study (cf. Section 7), layer L3 has been further subdivided to allow for a more strict separation between general legal and world knowledge (legal concepts and norms), cf. Section 7.1, from its *application* to relevant facts in the process of deciding a case (factual/contextual knowledge), cf. Section 7.2.

[5] The authors judiciously quote McCarty [62]: "The task for a lawyer or a judge in a 'hard case' is to construct a theory of the disputed rules that produces the desired legal result, and then to persuade the relevant audience that this theory is preferable to any theories offered by an opponent."

[6] In this article, we will actually associate type ι later on (cf. Section 5.2) with the domain of possible states/worlds.

[7] Note that functions of more than one argument can be represented in HOL in terms of functions of one argument. In this case, the values of these one-argument function applications are themselves functions, which are subsequently applied to the next argument. This technique, introduced by Schönfinkel [72], is commonly called *currying*; cf. Benzmüller and Andrews [58].

[8] HOL formulas (layer L0) should not be confused with the object-logical formulas (layer L1); the latter will later be identified in Section 5.2 with HOL predicates of type $\iota \to o$.

[9] For the purposes of the application scenarios studied later in Section 7, we have focused on $\mathcal{PL}$'s basic modal preference language, not yet employing ceteris paribus clauses. Nevertheless, we have provided a complete encoding and assessment of full $\mathcal{PL}$ in the associated *Isabelle/HOL* sources.

[10] Von Wright's proposal is discussed in some detail in van Benthem et al. [15]; cf. also Liu [80] for a discussion of further proposals.

[11] This corresponds to the well-known standard translation to first-order logic. Observe, however, that the additional expressivity of HOL allows us to also encode and flexibly combine non-normal modal logics (conditional, deontic, etc.; cf. [81–84]) and we can elegantly add quantifiers (cf. Section 5.4).

[12] In HOL (with Henkin semantics), sets are associated with their characteristic functions.

[13] For many logics, our embedding technique allows such faithfulness results, and an example for a non-trivial dyadic deontic logic has been worked out in detail by Benzmüller et al. [84]. A key feature of LogiKEy is that these pen-and-paper faithfulness studies can be supported and complemented by experimentation and verification. This is illustrated in Figure 1.2 of Benzmüller et al. [84], where it is shown that both the standard inference rules and the axioms of the embedded dyadic deontic logic can be proven valid in our approach. For further related experiments and a discussion of the current limitations of the embedding approach, we refer the reader to Benzmüller and Reiche [85] and in particular to §4.5 of Parent and Benzmüller [86], which briefly discusses in this context the distinction between validity on frame structures versus validity on model structures for a frame. An example of a non-trivial and non-faithful, but practically very successful, embedding in HOL is presented in the PhD thesis of Kirchner [87], where it is shown how to provide an appropriate additional safety harness.

[14] In fact, Halpern's [79] variant corresponds to employing the preference relation $\prec_{AE}$ discussed previously, augmented with an additional constraint to cope with infinite-sized countermodels to irreflexivity (building upon an approach by Lewis [89]). Thus, $\psi \succ^s \varphi$ (read: ψ is more likely than φ) if and only if every φ-state has a more likely ψ-state, say v, which *dominates* φ (i.e., no φ-state is more likely than v). Halpern [79] goes on to define a conditional operator as follows: $\varphi \Rightarrow \psi := \boldsymbol{A}\neg\varphi \vee ((\varphi \wedge \psi) \succ^s (\varphi \wedge \neg\psi))$.

[15] Observe that in doing so, we are simplifying the legal theory (*discoursive grammar*) to the effect that, for example, STAB*ility* becomes identified with EFFI*ciency*. This simplified model has proven sufficient for our modelling work in Section 7. A more granular encoding of principles is possible by adding a third axis to the value space in Figure 4, thus allocating each principle to its own octant

[16] We recall that, from a modal logic perspective, a proposition is modelled as the set of 'worlds' (i.e., states or situations) in which it holds. Informally, we want to be able to express the fact that a given principle, say legal STABility, is being observed or respected in a particular situation, or, abusing modal logic jargon, that the principle is 'satisfied' in that 'world'. This can become further interpreted as providing a *justification* for why that world or situation is desirable.

[17] An old mathematician's trick has been to employ—maybe unknowingly—Galois connections (respectively, adjunctions) to relate two universes of mathematical objects with each other in such a way that certain order structures become inverted (respectively, preserved). In doing so, insights and results can be transferred from a well-known universe towards a less-known one in order to gain information and help illuminate difficult problems; cf. the discussion in Erné [92].

[18] In particular, we want to highlight the potential of employing the powerful FCA methods, e.g., *attribute exploration* (Ganter et al. [94]), to prospective 'legal value mining' applications.

[19] The terms *extent* and *intent* are reminiscent of the philosophical notions of *extension* and *intension (comprehension)* reaching back to the 17th century *Logique de Port-Royal*.

[20] This result can be seamlessly stated for infinite meets and joins (infima and suprema) in the usual way. It corresponds to the first part of the so-called *basic theorem on concept lattices* (Ganter and Wille [93]).

[21] In the presented modelling, we intentionally avoided connecting our formal contexts with the betterness relation. This approach allows us to study the extent of our progress without it initially. Establishing such a connection in future work is possible, and LogiKEy is particularly well suited to supporting such logical explorations. One reviewer suggested making such a connection, and along the same lines another recommended representing value principles by Kripke relations and associated modal operators, since considering values as distinct modalities could naturally lead to their aggregation, potentially negating the need to define a separate aggregation function.

[22] Observe that this can be written semi-formally as: *for all w in $\mathcal{M}$ we have that if $\mathcal{M}, w \models A$ then $\mathcal{M}, w \models P{\downarrow}$*, which can be interpreted as "P provides a reason for all those worlds that satisfy A".

[23] Employing *discoursive grammar*'s value theory, this corresponds to RELIance together with personal GAIN outweighing STABility.

[24] Lacking any strong opinion regarding the correctness or adequacy of transitivity or the union property, we have nevertheless chosen this latter variant for our case study in Section 7, since it offers several benefits for our current modelling purposes: it can be faithfully encoded in the language of $\mathcal{PL}$ (van Benthem et al. [15]) and its behaviour is well documented in the literature; cf. Halpern [79], Liu [80] (Ch. 4). In fact, as mentioned in Section 5.4, drawing upon the strict variant $\prec_{AE}$ we can even define a defeasible conditional $\Rightarrow$ in $\mathcal{PL}$. Our choice of $\prec_{AE}/\preceq_{AE}$ thus strikes a good balance, it satisfies the desired properties, and it is used in related works (e.g., in preferential semantics for defeasible logics).

[25] Respective tests are presented in Figures A6 and A7 in Appendix A.1.

[26] We adopt the terminology of *promoting* (or *advancing*) a value from the literature (Bench-Capon and Sartor [5], Berman and Hafner [6], Prakken [51]) understanding it in a teleological sense: a decision promoting a value principle means taking that decision *for the sake* of observing the principle, thus seeing the value principle *as a reason* for making that decision.

[27] We shall not be held responsible for damages resulting from sloppy Latin paraphrasings!

[28] Recall our discussion in Section 6 (cf. note 15). In a future modelling of a (suitably enhanced) *discoursive grammar* (Section 2), we might take into account the order of combination of basic values in forming value principles, to the effect that, e.g., STABility can be properly distinguished from EFFIciency.

[29] *Nitpick* (Blanchette and Nipkow [95]) searches for, or, respectively, enumerates finite models or countermodels to a conjectured statement/lemma. By default, *Nitpick* searches for countermodels, and model finding is enforced by stating the parameter keyword 'satisfy'. These models are given as concrete interpretations of relevant terms in the given context so that the conjectured statement is satisfied or falsified.

[30] Further related tests are reported in Figure A9 in Appendix A.2.

[31] Cf. Berman and Hafner [6], Prakken [51], Bench-Capon [96], and also Gordon and Walton [97] for the significance of the Pierson v. Post case as a benchmark.

[32] Isabelle documents are suggestively called "theories". They correspond to top-level modules bundling together related definitions, theories, proofs, etc.

[33] Remember that a defeasible conditional implication can be defined employing $\mathcal{PL}$ modal operators; cf. Section 5.4. Alternatively we may also opt for an SSE of a conditional logic in HOL using other approaches as in Benzmüller [99].

[34] We use of the term 'default' in the colloquial sense of 'fallback', noting however, that there exist in fact several (non-monotonic) logical systems aimed at modelling such a kind of *defeasible*, aka. "default", behaviour for rules/conditionals (i.e., meaning that they can be 'overruled'). One of them has been suggestively called "default logic". We refer to Koons [100] for a discussion. In fact, and in the spirit of LogiKEy, we could have also employed, for encoding these rules, a $\mathcal{PL}$-defined defeasible conditional as discussed in Section 5.4. For the illustrative purposes of the present paper, and in view of the good performance of our present modelling, we did not yet find this step necessary.

[35] The introduction of legal *factors* is an established practice in the implementation of case-based legal systems (cf. Bench-Capon [56] for an overview). They can be conceived—as we do—as propositions abstracted from the facts of a case by the analyst/modeller in order to allow for assessing and comparing cases at a higher level of abstraction. Factors are typically either pro-plaintiff or pro-defendant, and their being true or false (respectively, present or absent) in a concrete case can serve to invoke relevant precedents or statutes.

[36] We note that our normative assignment here is widely in accordance with classifications in the AI and Law literature (Berman and Hafner [6], Bench-Capon [52]).

[37] The entire formalisation of this argument is presented in Figure A11 in Appendix A.4.

[38] Also observe that the legal precedent rule R4 of Keeble v. Hickeringill (see Figure A10, Line 39) as appears in Section 7.1 does not apply to this case.

[39] See the complete modelling in Figure A12 in Appendix A.4.

[40] The full details of the encoding are presented in Figure A13 in Appendix A.5.

[41] LogiKEy takes the position that expressive logics such as *classical* HOL (or possibly beyond, see Rothgang et al. [108]) are suited to serve as a universal meta-logic for knowledge representation and reasoning as motivated in Benzmüller [10] (regarding the choice of a meta-logic, we are thus in opposition to Quine [109], who advocated first-order logic for the task). This contrasts with the widespread view in the field of knowledge representation and reasoning in AI that decidability should be taken as a hard

limiting criterion for the development of any logic tools and associated infrastructure. We strongly disagree with this latter view for the following reasons. (i) Although our metalogic HOL is not decidable in general, many of its fragments, such as the guarded fragment, are. For example, after unfolding the formula $\Box\neg(A \land \neg A)$ using our embedding from Section 5, we obtain a decidable first-order formula in the guarded fragment, which can be effectively decided by various specialised and non-specialised tools in our LogiKEy infrastructure (and more reasoners can easily be added). (ii) What counts in applications is practical performance, not theoretical pen-and-paper results that often do not even find their way into implemented code. Especially for non-experts, there is hardly any difference between a decision procedure timing out because a problem is still too hard to solve within a given time limit, and a HOL prover giving up for the same reasons. (Remark on the performance of our HOL-based LogiKEy approach: verifying all of our example files as presented and discussed in the Appendix takes only 62 s of wall clock time and 76 s of cpu time on a Apple MacBook Pro (2019) with a 2.6 GHz 6-core Intel Core i7 processor and 16 GB of 2667 MHz DDR4 memory, and proving the decisions for Pierson or Post under their respective assumptions with *Sledgehammer*, when no clues are given, takes much less than 10 s) (iii) For the metalogical exploration studies presented in this paper, the use of an expressive metalogic based on the simply typed lambda calculus has been crucial. The widespread rejection of the simply typed lambda calculus in the area of knowledge representation and reasoning in AI seems counterintuitive anyway, given that it serves as the very foundation of all functional programming. (iv) To solve really hard and interesting problems, e.g., in mathematics, expressiveness can be crucial. HOL can allow hyperexponentially shorter proofs than are achievable in its decidable fragments so that some really interesting proofs are generally inaccessible in traditional (decidable) knowledge representation and reasoning frameworks, while they are not in our HOL-based approach; this has been demonstrated in [110].

References

1. Teubner, G. Substantive and Reflexive Elements in Modern Law. *Law Soc. Rev.* **1983**, *17*, 239–285. [CrossRef]
2. Lomfeld, B. Vor den Fällen: Methoden soziologischer Jurisprudenz. In *Die Fälle der Gesellschaft: Eine neue Praxis Soziologischer Jurisprudenz*; Lomfeld, B., Ed.; Mohr Siebeck: Tübingen, Germany, 2017; pp. 1–16.
3. Benzmüller, C.; Parent, X.; van der Torre, L. Designing Normative Theories for Ethical and Legal Reasoning: LogiKEy Framework, Methodology, and Tool Support. *Artif. Intell.* **2020**, *287*, 103348. [CrossRef]
4. Lomfeld, B. Grammatik der Rechtfertigung: Eine kritische Rekonstruktion der Rechts(fort)bildung. *Krit. Justiz* **2019**, *52*, 516–527. [CrossRef]
5. Bench-Capon, T.; Sartor, G. A model of legal reasoning with cases incorporating theories and value. *Artif. Intell.* **2003**, *150*, 97–143. [CrossRef]
6. Berman, D.; Hafner, C. Representing teleological structure in case-based legal reasoning: The missing link. In Proceedings of the 4th International Conference on Artificial Intelligence and Law, Amsterdam The Netherlands, 15–18 June 1993; ACM Press: New York, NY, USA, 1993; pp. 50–59.
7. Merrill, T.W.; Smith, H.E. *Property: Principles and Policies*; Foundation Press: Santa Barbara, CA, USA, 2017.
8. Casanovas, P.; Palmirani, M.; Peroni, S.; van Engers, T.M.; Vitali, F. Semantic Web for the Legal Domain: The next step. *Semant. Web* **2016**, *7*, 213–227. [CrossRef]
9. Hoekstra, R.; Breuker, J.; Bello, M.D.; Boer, A. LKIF Core: Principled Ontology Development for the Legal Domain. In *Law, Ontologies and the Semantic Web—Channelling the Legal Information Flood*; Frontiers in Artificial Intelligence and Applications; Breuker, J., Casanovas, P., Klein, M.C.A., Francesconi, E., Eds.; IOS Press: Amsterdam, The Netherlands, 2009; Volume 188, pp. 21–52. [CrossRef]
10. Benzmüller, C. Universal (Meta-)Logical Reasoning: Recent Successes. *Sci. Comput. Program.* **2019**, *172*, 48–62. [CrossRef]
11. Moor, J. Four kinds of ethical robots. *Philos. Now* **2009**, *72*, 12–14.
12. Scheutz, M. The Case for Explicit Ethical Agents. *AI Mag.* **2017**, *38*, 57–64. [CrossRef]
13. Arkin, R.C.; Ulam, P.; Duncan, B.A. *An Ethical Governor for Constraining Lethal Action in an Autonomous System*; Technical Report GVU-09-02; Georgia Institute of Technology: Atlanta, GA, USA, 2009.
14. Benzmüller, C.; Fuenmayor, D. Value-oriented Legal Argumentation in Isabelle/HOL. In *International Conference on Interactive Theorem Proving (ITP), Proceedings*; LIPIcs; Cohen, L., Kaliszyk, C., Eds.; Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Wadern, Germany, 2021; Volume 193, pp. 23:1–23:18. [CrossRef]
15. van Benthem, J.; Girard, P.; Roy, O. Everything Else Being Equal: A Modal Logic for *Ceteris Paribus* Prefer. *J. Philos. Log.* **2009**, *38*, 83–125. [CrossRef]
16. Prakken, H.; Sartor, G. Law and logic: A review from an argumentation perspective. *Artif. Intell.* **2015**, *227*, 214–225. [CrossRef]
17. Alexy, R. (Ed.) *Theorie der juristischen Argumentation*; Suhrkamp: Frankfurt, Germany, 1978.
18. Feteris, E. *Fundamentals of Legal Argumentation*; Springer: Dordrecht, The Netherlands, 2017.
19. Hage, J. *Reasoning with Rules*; Kluwer: Dordrecht, The Netherlands, 1997.
20. Prakken, H. *Logical Tools for Modelling Legal Argument*; Springer: Dordrecht, The Netherlands, 1997.
21. Modgil, S.; Prakken, H. Abstract Rule-Based Argumentation. In *Handbook of Formal Argumentation*; Baroni, P., Gabbay, D., Giacomin, M., van der Torre, L., Eds.; College Publications: Rickmansworth, UK, 2018; pp. 287–364.
22. Ashley, K.D. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*; MIT Press: Cambridge, MA, USA, 1990.
23. Aleven, V. Teaching Case-Based Reasoning through a Model and Examples. Ph.D. Dissertation, University of Pittsburgh, Pittsburgh, PA, USA, 1997.

24. Horty, J. Rules and reasons in the theory of precedent. *Leg. Theory* **2011**, *17*, 1–33. [CrossRef]
25. Bench-Capon, T.; Atkinson, K.; Chorley, A. Persuasion and value in legal argument. *J. Log. Comput.* **2005**, *15*, 1075–1097. [CrossRef]
26. Grabmair, M. Modeling Purposive Legal Argumentation and Case Outcome Prediction Using Argument Schemes in the Value Judgment Formalism. Ph.D. Dissertation, University of Pittsburgh, Pittsburgh, PA, USA, 2016.
27. Maranhão, J.; Sartor, G. Value assessment and revision in legal interpretation. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, ICAIL 2019, Montreal, QC, Canada, 17–21 June 2019; pp. 219–223. [CrossRef]
28. Lomfeld, B. *Die Gründe des Vertrages: Eine Diskurstheorie der Vertragsrechte*; Mohr Siebeck: Tübingen, Germany, 2015.
29. Alexy, R. On Balancing and Subsumption: A Structural Comparison. *Ratio Juris* **2003**, *16*, 433–449. [CrossRef]
30. Sartor, G. Doing justice to rights and values: Teleological reasoning and proportionality. *Artif. Intell. Law* **2010**, *18*, 175–215. [CrossRef]
31. Sartor, G. A Quantitative Approach to Proportionality. In *Handbook of Legal Reasoning and Argumentation*; Bongiovanni, G., Postema, G., Rotolo, A., Sartor, G., Valentini, C., Walton, D., Eds., Springer: Dordrecht, The Netherlands, 2018; pp. 613–636.
32. Dworkin, R. *Taking Rights Seriously*; Harvard University Press: Cambridge, MA, USA, 1978; OCLC: 4313351.
33. Alexy, R. On the Structure of Legal Principles. *Ratio Juris* **2000**, *13*, 294–304. [CrossRef]
34. Raz, J. Legal Principles and the Limits of Law. *Yale Law J.* **1972**, *81*, 823–854. [CrossRef]
35. Verheij, B.; Hage, J.C.; Van Den Herik, H.J. An integrated view on rules and principles. *Artif. Intell. Law* **1998**, *6*, 3–26. [CrossRef]
36. Neves, M. *Constitutionalism and the Paradox of Principles and Rules*; Oxford University Press: Oxford, UK, 2021.
37. Barak, A. *Proportionality*; Cambridge University Press: Cambridge, UK, 2012.
38. van der Weide, T.; Dignum, F.; Meyer, J.-J.C.; Prakken, H.; Vreeswijk, G. Practical Reasoning Using Values. In *Argumentation in Multi-Agent Systems (ArgMAS)*; McBurney, P., Rahwan, I., Parsons, S., Maudet, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 79–93.
39. Gruber, T. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.* **1993**, *5*, 199–220. [CrossRef]
40. Gruber, T. Ontology. In *Encyclopedia of Database Systems*; Liu, L., Özsu, M.T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009.
41. Smith, B. Ontology. In *Blackwell Guide to the Philosophy of Computing and Information*; Floridi, L., Ed.; Blackwell: Oxford, UK, 2003.
42. Rokeach, M. *The Nature of Human Values*; Free Press Macmillan: New York, NY, USA, 1973.
43. Schwartz, S. Universals in the Content and Structure of Values. *Adv. Exp. Soc. Psychol.* **1992**, *25*, 1–65.
44. Eysenck, H. *The Psychology of Politics*; Routledge: London, UK, 1954.
45. Mitchell, B. *Eight Ways to Run the Country*; Praeger: Westport, CT, USA, 2007.
46. Clark, B. *Political Economy: A Comparative Approach*; Praeger: New York, NY, USA, 1991.
47. Hofstede, G. *Culture's Consequences*; Sage: Thousands Oaks, CA, USA, 2001.
48. Inglehart, R. *Cultural Evolution*; Cambridge University Press: Cambridge, CA, USA, 2018.
49. Bench-Capon, T. Ethical approaches and autonomous systems. *Artif. Intell.* **2020**, *281*, 103239.[CrossRef]
50. Sartor, G. Teleological arguments and theory-based dialectics. *Artif. Intell. Law* **2002**, *10*, 95–112. [CrossRef]
51. Prakken, H. An exercise in formalising teleological case-based reasoning. *Artif. Intell. Law* **2002**, *10*, 113–133. [CrossRef]
52. Bench-Capon, T. Representing Popov v Hayashi with dimensions and factors. *Artif. Intell. Law* **2012**, *20*, 15–35. [CrossRef]
53. Gordon, T.; Walton, D. A Carneades reconstruction of Popov v Hayashi. *Artif. Intell. Law* **2012**, *20*, 37–56. [CrossRef]
54. Bench-Capon, T.; Prakken, H. Using argument schemes for hypothetical reasoning in law. *Artif. Intell. Law* **2010**, *18*, 153–174. [CrossRef]
55. Chorley, A.; Bench-Capon, T. An empirical investigation of reasoning with legal cases through theory construction and application. *Artif. Intell. Law* **2005**, *13*, 323–371. [CrossRef]
56. Bench-Capon, T. Hypo's legacy: Introduction to the virtual special issue. *Artif. Intell. Law* **2017**, *25*, 205–250. [CrossRef]
57. Verheij, B. Formalizing value-guided argumentation for ethical systems design. *Artif. Intell. Law* **2016**, *24*, 387–407. [CrossRef]
58. Benzmüller, C.; Andrews, P. Church's Type Theory. In *The Stanford Encyclopedia of Philosophy*, Summer 2019 ed.; Zalta, E.N., Ed.; Metaphysics Research Lab, Stanford University: Stanford, CA, USA, 2019; pp. 1–62.
59. Gibbons, J.; Wu, N. Folding domain-specific languages: Deep and shallow embeddings (functional Pearl). In Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming, Gothenburg, Sweden, 1–3 September 2014; Jeuring, J., Chakravarty, M.M.T., Eds.; ACM: New York, NY, USA, 2014; pp. 339–347. [CrossRef]
60. Svenningsson, J.; Axelsson, E. Combining Deep and Shallow Embedding for EDSL. In *Trends in Functional Programming*; Loidl, H.W., Peña, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 21–36.
61. Blanchette, J.C.; Kaliszyk, C.; Paulson, L.C.; Urban, J. Hammering towards QED. *J. Formaliz. Reason.* **2016**, *9*, 101–148.
62. McCarty, L.T. An implementation of Eisner v. Macomber. In Proceedings of the 5th International Conference on Artificial Intelligence and Law, College Park, MD, USA, 21–24 May 1995; pp. 276–286.
63. Fuenmayor, D.; Benzmüller, C. A Computational-Hermeneutic Approach for Conceptual Explicitation. In *Model-Based Reasoning in Science and Technology. Inferential Models for Logic, Language, Cognition and Computation*; SAPERE; Nepomuceno, A., Magnani, L., Salguero, F., Bares, C., Fontaine, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 49, pp. 441–469. [CrossRef]
64. Daniels, N. Reflective Equilibrium. In *The Stanford Encyclopedia of Philosophy*, Summer 2020 ed.; Zalta, E.N., Ed.; Metaphysics Research Lab, Stanford University: Stanford, CA, USA, 2020.
65. Rawls, J. *A Theory of Justice*; Harvard University Press: Cambridge, MA, USA, 1971; Revised edition 1999.

66. Goodman, N. *Fact, Fiction, and Forecast*; Harvard University Press: Cambridge, MA, USA, 1955.
67. Andrews, P.B. General Models, Descriptions, and Choice in Type Theory. *J. Symb. Log.* **1972**, *37*, 385–394. [CrossRef]
68. Andrews, P.B. General Models and Extensionality. *J. Symb. Log.* **1972**, *37*, 395–397. [CrossRef]
69. Benzmüller, C.; Brown, C.; Kohlhase, M. Higher-Order Semantics and Extensionality. *J. Symb. Log.* **2004**, *69*, 1027–1088. [CrossRef]
70. Benzmüller, C.; Miller, D. Automation of Higher-Order Logic. In *Handbook of the History of Logic, Volume 9—Computational Logic*; Gabbay, D.M., Siekmann, J.H., Woods, J., Eds.; Elsevier: North Holland, The Netherlands, 2014; pp. 215–254. [CrossRef]
71. Church, A. A Formulation of the Simple Theory of Types. *J. Symb. Log.* **1940**, *5*, 56–68. [CrossRef]
72. Schönfinkel, M. Über die Bausteine der mathematischen Logik. *Math. Ann.* **1924**, *92*, 305–316. [CrossRef]
73. Henkin, L. Completeness in the Theory of Types. *J. Symb. Log.* **1950**, *15*, 81–91. [CrossRef]
74. von Wright, G.H. *The Logic of Preference*; Edinburgh University Press: Edinburgh, UK, 1963.
75. Benzmüller, C.; Paulson, L.C. Multimodal and Intuitionistic Logics in Simple Type Theory. *Log. J. IGPL* **2010**, *18*, 881–892. [CrossRef]
76. Benzmüller, C.; Paulson, L.C. Quantified Multimodal Logics in Simple Type Theory. *Log. Universalis* **2013**, *7*, 7–20. [CrossRef]
77. Carnielli, W.; Coniglio, M.E. Combining Logics. In *The Stanford Encyclopedia of Philosophy*, Fall 2020 ed.; Zalta, E.N., Ed.; Metaphysics Research Lab, Stanford University: Stanford, CA, USA, 2020.
78. Carnielli, W.; Coniglio, M.; Gabbay, D.M.; Paula, G.; Sernadas, C. *Analysis and Synthesis of Logics*; Number 35 in Applied Logics Series; Springer: Berlin/Heidelberg, Germany, 2008.
79. Halpern, J.Y. Defining relative likelihood in partially-ordered preferential structures. *J. Artif. Intell. Res.* **1997**, *7*, 1–24. [CrossRef]
80. Liu, F. Changing for the Better: Preference Dynamics and Agent Diversity. Ph.D. Thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam: Amsterdam, The Netherlands, 2008.
81. Benzmüller, C. Cut-Elimination for Quantified Conditional Logic. *J. Philos. Log.* **2017**, *46*, 333–353. [CrossRef]
82. Benzmüller, C.; Farjami, A.; Meder, P.; Parent, X. I/O Logic in HOL. *J. Appl. Logics—IfCoLoG J. Logics Their Appl.* **2019**, *6*, 715–732.
83. Benzmüller, C.; Farjami, A.; Parent, X. Åqvist's Dyadic Deontic Logic E in HOL. *J. Appl. Logics—IfCoLoG J. Logics Their Appl.* **2019**, *6*, 733–755.
84. Benzmüller, C.; Farjami, A.; Parent, X. Dyadic Deontic Logic in HOL: Faithful Embedding and Meta-Theoretical Experiments. In *New Developments in Legal Reasoning and Logic: From Ancient Law to Modern Legal Systems*; Logic, Argumentation & Reasoning; Rahman, S., Armgardt, M., Kvernenes, N., Christian, H., Eds.; Springer Nature: Cham, Switzerland, 2022; Volume 23. [CrossRef]
85. Benzmüller, C.; Reiche, S. Automating Public Announcement Logic with Relativized Common Knowledge as a Fragment of HOL in LogiKEy. *J. Log. Comput.* **2022**, *33*, 1243–1269. [CrossRef]
86. Parent, X.; Benzmüller, C. Normative conditional reasoning as a fragment of HOL. *arXiv Preprint* **2024**, arXiv:2308.10686. https://doi.org/10.48550/arXiv.2308.10686
87. Kirchner, D. Computer-Verified Foundations of Metaphysics and an Ontology of Natural Numbers in Isabelle/HOL. Ph.D. Thesis, Freie Universität Berlin, Berlin, Germany, 2022. [CrossRef]
88. Boutilier, C. Toward a logic for qualitative decision theory. In *Principles of Knowledge Representation and Reasoning*; Elsevier: Amsterdam, The Netherlands, 1994; pp. 75–86. [CrossRef]
89. Lewis, D. *Counterfactuals*; Harvard University Press: Cambridge, MA, USA, 1973.
90. Van Benthem, J. For Better or for Worse: Dynamic Logics of Preference. In *Preference Change: Approaches from Philosophy, Economics and Psychology*; Grüne-Yanoff, T., Hansson, S.O., Eds.; Springer: Dordrecht, The Netherlands, 2009; pp. 57–84. [CrossRef]
91. Liu, F. *Reasoning about Preference Dynamics*; Springer: Dordrecht, The Netherlands, 2011. [CrossRef]
92. Erné, M. Adjunctions and Galois Connections: Origins, History and Development. In *Galois Connections and Applications*; Denecke, K., Erné, M., Wismath, S.L., Eds.; Springer: Dordrecht, The Netherlands, 2004; pp. 1–138. [CrossRef]
93. Ganter, B.; Wille, R. *Formal Concept Analysis: Mathematical Foundations*; Springer: Berlin/Heidelberg, Germany, 2012.
94. Ganter, B.; Obiedkov, S.; Rudolph, S.; Stumme, G. *Conceptual Exploration*; Springer: Berlin/Heidelberg, Germany, 2016.
95. Blanchette, J.C.; Nipkow, T. Nitpick: A Counterexample Generator for Higher-Order Logic Based on a Relational Model Finder. In *Interactive Theorem Proving, Proceedings of the First International Conference, ITP 2010, Edinburgh, UK, 11–14 July 2010;* LNCS; Kaufmann, M., Paulson, L.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6172, pp. 131–146.
96. Bench-Capon, T. The missing link revisited: The role of teleology in representing legal argument. *Artif. Intell. Law* **2002**, *10*, 79–94. [CrossRef]
97. Gordon, T.F.; Walton, D. Pierson vs. Post revisited. *Front. Artif. Intell. Appl.* **2006**, *144*, 208.
98. Blanchette, J.C.; Böhme, S.; Paulson, L.C. Extending Sledgehammer with SMT Solvers. *J. Autom. Reason.* **2013**, *51*, 109–128. [CrossRef]
99. Benzmüller, C. Automating Quantified Conditional Logics in HOL. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-13), Beijing, China, 3–9 August 2013; pp. 746–753.
100. Koons, R. Defeasible Reasoning. In *The Stanford Encyclopedia of Philosophy*, Winter 2017 ed.; Zalta, E.N., Ed.; Metaphysics Research Lab, Stanford University: Stanford, CA, USA, 2017.
101. Wenzel, M. Isabelle/Isar—A generic framework for human-readable proof documents. *Insight Proof-Festschr. Honour Andrzej Trybulec* **2007**, *10*, 277–298.
102. Rissland, E.L.; Ashley, K.D. A case-based system for trade secrets law. In Proceedings of the 1st International Conference on Artificial Intelligence and Law, Boston, MA, USA, 27 May 1997–29 May 1987; pp. 60–66.

103. Krause, P.; Ambler, S.; Elvang-Goransson, M.; Fox, J. A Logic Of Argumentation for Reasoning under Uncertainty. *Comput. Intell.* **1995**, *11*, 113–131. [CrossRef]
104. Carnielli, W.; Coniglio, M.E.; Fuenmayor, D. Logics of Formal Inconsistency Enriched with Replacement: An Algebraic and Modal Account. *Rev. Symb. Log.* **2021**, *15*, 771–806. [CrossRef]
105. Fuenmayor, D. Topological Semantics for Paraconsistent and Paracomplete Logics. Archive of Formal Proofs. 2020. Available online: https://isa-afp.org/entries/Topological_Semantics.html (accessed on 12 December 2023).
106. Benzmüller, C.; Lomfeld, B. Reasonable Machines: A Research Manifesto. In *KI 2020: Advances in Artificial Intelligence, Proceedings of the 43rd German Conference on Artificial Intelligence, Bamberg, Germany, 21–25 September 2020*; Lecture Notes in Artificial Intelligence; Schmid, U., Klügl, F., Wolter, D., Eds.; Springer: Cham, Switzerland, 2020; Volume 12352, pp. 251–258. [CrossRef]
107. Fuenmayor, D.; Benzmüller, C. Normative Reasoning with Expressive Logic Combinations. In *ECAI 2020, Proceedings of the 24th European Conference on Artificial Intelligence, Santiago de Compostela, Spain, 8–12 June 2020*; Frontiers in Artificial Intelligence and Applications; De Giacomo, G., Catala, A., Dilkina, B., Milano, M., Barro, S., Bugarín, A., Lang, J., Eds.; IOS Press: Amsterdam, The Netherlands, 2020; Volume 325, pp. 2903–2904. [CrossRef]
108. Rothgang, C.; Rabe, F.; Benzmüller, C. Theorem Proving in Dependently-Typed Higher-Order Logic. In *Automated Deduction—CADE 29, Proceedings of the 29th International Conference on Automated Deduction, Rome, Italy, 1–4 July 2023;* Lecture Notes in Artificial Intelligence; Pientka, B., Tinelli, C., Eds.; Springer: Cham, Switzerland, 2023; Volume 14132, pp. 438–455. [CrossRef]
109. Hylton, P.; Kemp, G. Willard Van Orman Quine. In *The Stanford Encyclopedia of Philosophy*, Fall 2023 ed.; Zalta, E.N., Nodelman, U., Eds.; Metaphysics Research Lab, Stanford University: Stanford, CA, USA, 2023.
110. Benzmüller, C.; Fuenmayor, D.; Steen, A.; Sutcliffe, G. Who Finds the Short Proof? *Log. J. IGPL* **2023.** [CrossRef]
111. Bench-Capon, T. The Need for Good Old-Fashioned AI and Law. In *International Trends in Legal Informatics: A Festschrift for Erich Schweighofer*; Hötzendorfer, W., Tschol, C., Kummer, F., Eds.; Weblaw: Bern, Switzerland, 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.