Encoding Legal Balancing: Automating an Abstract Ethico-Legal Value Ontology in Preference Logic

Christoph Benzmüller[1] , **David Fuenmayor**[1] , **Bertram Lomfeld**[2]
[1]Department of Computer Science and Mathematics, FU Berlin, Berlin, Germany
[2]Department of Law, FU Berlin, Berlin, Germany
{c.benzmueller, david.fuenmayor, bertram.lomfeld}@fu-berlin.de

Abstract

Enabling machines to legal balancing is a non-trivial task challenged by a multitude of factors some of which are addressed and explored in this work. We propose a holistic approach to formal modeling at different abstraction layers supported by a pluralistic framework in which the encoding of an ethico-legal value and upper ontology is developed in combination with the exploration of a formalization logic, with legal domain knowledge and with exemplary use cases until a reflective equilibrium is reached. Our work is enabled by a meta-logical approach to universal logical reasoning and it applies the recently introduced *LogiKEy* methodology for designing normative theories for ethical and legal reasoning. The particular focus in this paper is on the formalization and encoding of a value ontology suitable e.g. for explaining and resolving legal conflicts in property law (wild animal cases).

1 Introduction

Law today has to reflect highly pluralistic environments. There are plural values, world-views, and logics which may even be considered as constituting plural worlds. One function of modern, reflexive law is to enable the social interaction within and between such plural worlds (Teubner 1983; Lomfeld 2017). Any sound model of legal reasoning needs to be pluralistic while at the same time reflecting the uniting force of law.

Logical reconstructions of legal reasoning quite often separate deductive rule application and legal interpretation, cf. the overview in (Prakken and Sartor 2015). Yet, understanding legal reasoning as practical argumentation (Alexy 1978), this separation breaks down. Nonetheless different types of rule-based (Hage 1997), case-based (Horty 2011) and value-based (Bench-Capon et al. 2005) reasoning remain. A discourse theory of law could integrate these different types by translating legal reasoning into the balancing of plural and opposing socio-legal values (Lomfeld 2015).

There are several models to quantify legal balancing, e.g. (Alexy 2003; Sartor 2010). Nonetheless, these approaches need to get "integrated with logic and argumentation to provide a comprehensive account of value-based reasoning" (Sartor 2018). An adequate balancing model of legal reasoning, thus, has to reconstruct rule subsumption and case interpretation as argumentation process between conflicting values. Here, the differentiation of legal norms into rules and principles reveals its potential (Alexy 2000). Legal principles could be understood as material values on a deep level of legal balancing, which is structured by legal rules on an explicit upper level of legal reasoning (Lomfeld 2015).

Within the recent AI & Law perspective value conflicts are often modeled according to Dung's abstract argumentation (Dung 1995) as a value argumentation framework (Bench-Capon et al. 2005). Yet, if preferences between defeasible rules are reconstructed and justified in terms of preferences between values, some questions about values necessarily pop up: "Are values scalar? [...] Can values be ordered at all? [...] How can sets of values be compared? [...] Can several less important values together overcome a more important value?" (Bench-Capon and Sartor 2003).

Thus, an encompassing approach for legal reasoning as practical argumentation needs not only a formal reconstruction of the relation between legal values (principles) and legal rules, but also a value ontology that allows to systematize value comparison and conflicts as "discoursive grammar" (Lomfeld 2015; Lomfeld 2019).

In this paper we demonstrate how an abstract level encoding and automation of an ethico-legal value ontology, i.e. Lomfeld's "discoursive grammar" of justification, can be achieved by reusing and extending the *LogiKEy* methodology and formal framework (Benzmüller et al. 2020). This framework works with shallow semantical embeddings (SSEs) of (combinations of) non-classical logics in classical higher-order logic (HOL). HOL thereby serves as a meta-logic, rich enough to support the encoding of a plurality of "object logics" (e.g. conditional, deontic or epistemic logics and combinations thereof) and plural and adaptable value systems. The embedded "object logics" are used for the iterative, experimental encoding of normative theories, which finally help to reason with concrete legal cases utilizing the encoded value system. This reflects ideas of understanding the solution of legal cases as theory construction, "building, evaluating and using theories" (Bench-Capon and Sartor 2003).[1] This multi-layered knowledge engineering process is supported in our framework by adapting automated

[1]They quote e.g. McCarty (1995, p285): "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."

theorem proving (ATP) technology for meta-logic HOL.

Ethico-legal ontologies also constitute a core ingredient to enable the computation, assessment and communication of abstract-level justifications in the future ethico-legal governance of AI (Benzmüller and Lomfeld 2020). A sound implementation of any legally accountable "moral machine" requires the development of upper-level ontologies to guide and connect the encoding of concrete regulatory codes (Hoekstra et al. 2009; Fuenmayor and Benzmüller 2019b) or legal cases. Understanding legal reasoning as conflictual practical argumentation, the plural interpretation of these concrete legal rules requires a complementary abstract ethico-legal value ontology, for example, as a "discoursive grammar" of justification (Lomfeld 2019).

The contributions of this paper are manifold: After summarizing Lomfeld's value ontology, resp. "discoursive grammar" of justification, in §2, and briefly depicting the *LogiKEy* development and knowledge engineering methodology in §3, we outline our object logic of choice – the preference logic by van Benthem et al. (2009) – in §4. In this section we then present, as our first main contribution, a semantical embedding of this preference logic in the *Isabelle/HOL* proof assistant (Nipkow et al. 2002), and we test and assess its meta-theory with state-of-the-art ATP technology. Subsequently we encode in §5, as our second main contribution, Lomfeld's value ontology, which is again extensively tested and assessed with ATP systems. Further relevant legal and world knowledge is modeled and added in §6, and then it is demonstrated in §7 how the introduced and formalized (general) knowledge can be used for automatically generating value-oriented computer explanations for exemplary property law ("wild animal") cases. §8 concludes the paper.

Formalization sources are available at http://logikey.org under "Preference-Logics/vanBenthemEtAl2009."

2 "Discoursive Grammar" of Justification

Combining the discourse theoretical idea that legal reasoning is practical argumentation with a two-level model of legal norms, legal "rules" could be reconstructed as conditional preference relations between conflicting underlying value "principles" (Alexy 2000; Lomfeld 2015). The legal consequence of a rule R implies the value preference of value principle A over value principle B: $A > B$ (e.g. health security outweighs freedom to move).[2] This value preference applies under the condition that the norm elements E_1 and E_2 which the rule denotes as prerequisites are met. Thus, if E_1 and E_2 are satisfied (e.g. a virus pandemic occurs and voluntary shut down fails), then $A > B$ which justifies the rule's legal consequence (e.g. sanctioned lock-down). We thus have a *conditional preference relation*

$$R : (E_1 \wedge E_2) \rightarrow A > B$$

But which value principles are to be balanced? How to find a suitable justification framework? Based on comparative discourse analyses in different legal systems, one could reconstruct a general dialectical taxonomy of legal value

[2] In §5 these values will be assigned to particular parties/actors, i.e. ruling for different parties may promote different values.

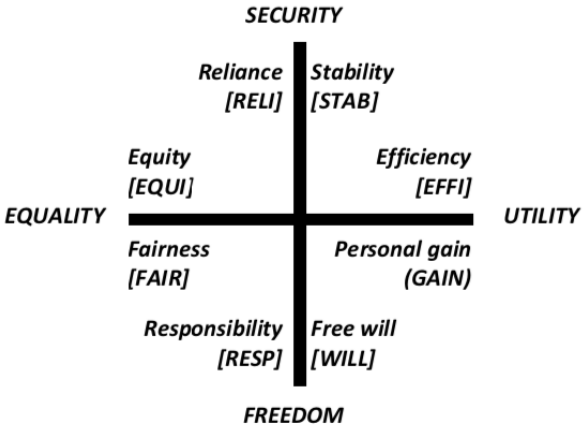

Figure 1: Value ontology by (Lomfeld 2019)

principles used in (at least Western) legislation, legislative materials, cases, textbooks and scholar writings (Lomfeld 2015). The idea is to provide a consistent systematic value ontology of legal principles independent of concrete cases or legal fields to justify legal decisions.

The proposed legal value ontology, see Fig. 1, is consistent with similar taxonomies of political and economic values (Lomfeld 2019). In all these social systems one could observe a general antinomy between individual and collective values. Ideal types of this basic dialectic are the value of FREEDOM for the individual and the value of SECURITY for the collective perspective. Another classic social value antinomy is between a functional (economic) and a more idealistic viewpoint, represented in the ethical debate by the value dialectic of UTILITY versus EQUALITY. These four normative poles stretch an axis of value coordinates for the general set construction.

Within this dialectical matrix eight more concrete legal values/principles are identified. FREEDOM represents the normative value of individual autonomy and comprises the legal principles of more functional individual choice or 'free will' (WILL) and more idealistic (self-)'responsibility' (RESP). The value of SECURITY addresses the collective dimension of public order and comprises the legal principles of 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 level and comprises the legal principles of collective overall welfare-maximization, i.e. 'efficiency' (EFFI) and individual welfare-maximization, 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).

This legal value ontology could consistently cover existing value sets from law & logic (or AI & Law) accounts on value-based reasoning, e.g. (Berman and Hafner 1993; Bench-Capon 2012; Gordon and Walton 2012; Sartor 2010), mostly modeled in connection with so-called "wild animal cases" or succeeding common law property cases.

3 Methodology

The *LogiKEy* methodology (Benzmüller et al. 2020) refers to the principles underlying the organization and the conduct of a complex knowledge design and engineering process—which is what we are faced with. Design means the depiction of the main features of the system we want to achieve, and (knowledge or logic) engineering refers to all the technical and scientific aspects involved in building, maintaining and using a knowledge-based, resp. logic-based, system.

LogiKEy's unifying formal framework is based on (shallow) semantical embeddings (SSEs) of 'object' logics into a higher-order meta-logic (HOL), which enables the provision of powerful tool support: off-the-shelf theorem provers and model finders for HOL are assisting the *LogiKEy* knowledge engineer to *flexibly experiment* with underlying logics and their combinations, with (legal & world) general and domain knowledge, and with concrete use cases—all at the same time. Continuous improvements of these off-the-shelf provers, without further ado, leverage the reasoning performance in *LogiKEy*.

In our context, the knowledge engineering task is facing the layers as depicted in Fig. 2: meta-logic HOL enables the encoding of a preference logic, followed by the value ontology, both of which are then used for formalizing legal & world knowledge, before concrete wild animal use cases are modeled and assessed.

The engineering process at these layers has backtracking points and several work cycles may be required; thereby the higher layers may also pose modification requests to the lower layers. Such requests may, unlike in most other approaches, also include far-reaching modifications of the chosen logical foundations, e.g. in the adopted particular preference logic. The work we present in this paper is in fact the result of an iterative, give-and-take process encompassing several cycles of modeling, assessment and testing activities, whereby a (modular) logical theory gradually evolves until eventually reaching a state of highest coherence and acceptability. We then speak of arriving at a *reflective equilibrium*, cf. previous work on *computational hermeneutics* (Fuenmayor and Benzmüller 2019a).

LogiKEy thus supports empirical studies on ethico-legal theories in which the underlying logic formalisms itself can be flexibly varied, assessed and compared in context.

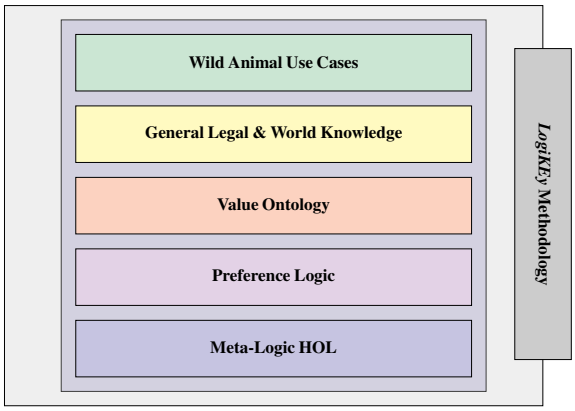

Figure 2: *LogiKEy* development methodology

4 Preference Logic

Adopting the *LogiKEy* methodology, the first question to address is: Which preference logic to choose (initially)? Preference logics generally aim at an adequate modeling of preference reasoning of an (idealized) rational individual. Hence, the logic we choose needs to provably support relevant criteria (e.g., strengthening/weakening the atecedent/consequent). Our initial choice has been the preference logic by van Benthem et al. (2009), abbreviated by $\mathcal{PL}$ in the remainder: $\mathcal{PL}$ is a *"modal logic for ceteris paribus preferences understood in the sense of 'all other things being equal'. This reading goes back to the seminal work of Von Wright in the early 1960's and has returned in computer science in the 1990's and in more abstract 'dependency logics' today."* $\mathcal{PL}$ appears suited for effective automation using the SSEs approach, which has been another selection criterion. This judgment is based on good prior experience with SSEs of related logics whose semantics employs accessibility relations between possible worlds/states, just as $\mathcal{PL}$ does. Interesting alternatives to $\mathcal{PL}$ include the preference logics by Liu (2008) and Osherson and Weinstein (2012).

$\mathcal{PL}$ may be criticized as an overly complex choice for our aims as presented in §7, where the automation of *ceteris paribus* reasoning has hardly been addressed yet (but will be in future work). Anyhow, we may always 'downgrade' our choice of logic if that is needed by e.g. practical performance considerations, or we may 'upgrade' or combine it with other logics if required. Remember that the choice of formalization logic is a parameter in the *LogiKEy* approach.

Automating Preference Logic $\mathcal{PL}$

For the mechanization and automation of $\mathcal{PL}$ we utilize the SSE technique, which encodes the language constituents of an object logic, $\mathcal{PL}$ in our case, as expressions (λ-terms) in HOL. This shows that $\mathcal{PL}$ is just a fragment of HOL and can be automated as such. A core idea is to model (relevant parts of) the semantic structures of $\mathcal{PL}$ explicitly in HOL. For $\mathcal{PL}$ these structures are relational frames constituted by sets of possible worlds/states and their accessibility relations. $\mathcal{PL}$ propositions can thus be encoded in the SSE as predicates in HOL taking possible worlds/states as arguments.

The SSE of the basic operators of preference logic $\mathcal{PL}$ in HOL is depicted in Fig. 3; this encoding is further explained below. Further extensions to support *ceteris paribus* reasoning in $\mathcal{PL}$ are presented in Fig. 9 in Appx. A.1.

As a result we obtain a combined, interactive and automated, theorem prover and model finder for $\mathcal{PL}$ realized within *Isabelle/HOL*. This is a new contribution, since we are not aware of any other existing implementation and automation of $\mathcal{PL}$. Moreover, as we will demonstrate below, the SSE technique supports the automated assessment of meta-logical properties of the embedded logic in *Isabelle/HOL*, which in turn provides practical evidence for the correctness of our encoding. A formal proof of the faithfulness (soundness & completeness) of the SSE is analogous to numerous prior works, cf. Benzmüller (2019) and the references therein.

```
theory PreferenceLogicBasics (** Benzmüller & Fuenmayor, 2020 **)
 imports Main
begin (** SSE of prefer. logic by van Benthem et al., JPL 2009 **)
(*unimportant*)declare[[syntax_ambiguity_warning=false]]
nitpick_params[user_axioms,expect=genuine,show_all,format=3]
(*preliminaries*)
typedecl i                    (*possible worlds*)
type_synonym σ="i⇒bool"       (*'world-lifted' propositions*)
type_synonym γ="i⇒i⇒bool"     (*preference relations*)
type_synonym μ="σ⇒σ"          (*unary logical connectives*)
type_synonym ν="σ⇒σ⇒σ"        (*binary logical connectives*)
type_synonym π="σ⇒bool"       (*sets of world-lifted propositions*)
(*betterness relation ⪯ and strict betterness relation ≺*)
consts BR::γ ("_⪯_")
abbreviation SBR::γ ("_≺_") where "v≺w ≡ (v⪯w) ∧ ¬(w⪯v)"
abbreviation "reflexive R ≡ ∀x. R x x"
abbreviation "transitive R ≡ ∀x y z. R x y ∧ R y z ⟶ R x z"
abbreviation "is_total R ≡ ∀x y. R x y ∨ R y x"
axiomatization where rBR: "reflexive BR" and tBR: "transitive BR"
(*modal logic connectives (operating on truth-sets)*)
abbreviation c1::σ ("⊥")   where "⊥ ≡ λw. False"
abbreviation c2::σ ("⊤")   where "⊤ ≡ λw. True"
abbreviation c3::μ ("¬_")  where "¬φ ≡ λw.¬(φ w)"
abbreviation c4::ν (infixl"∧"85) where "φ∧ψ ≡ λw.(φ w)∧(ψ w)"
abbreviation c5::ν (infixl"∨"83) where "φ∨ψ ≡ λw.(φ w)∨(ψ w)"
abbreviation c6::ν (infixl"→"84) where "φ→ψ ≡ λw.(φ w)⟶(ψ w)"
abbreviation c7::ν (infixl"↔"84) where "φ↔ψ ≡ λw.(φ w)⟷(ψ w)"
abbreviation c8::μ ("□⪯_") where "□⪯φ ≡ λw.∀v.(w⪯v)⟶(φ v)"
abbreviation c9::μ ("◇⪯_") where "◇⪯φ ≡ λw.∃v.(w⪯v)∧(φ v)"
abbreviation c10::μ ("□≺_") where "□≺φ ≡ λw.∀v.(w≺v)⟶(φ v)"
abbreviation c11::μ ("◇≺_") where "◇≺φ ≡ λw.∃v.(w≺v)∧(φ v)"
abbreviation c12::μ ("E_") where "Eφ ≡ λw.∃v.(φ v)"
abbreviation c13::μ ("A_") where "Aφ ≡ λw.∀v.(φ v)"
(*meta-logical predicate for global and validity*)
abbreviation g1::π ("⌊_⌋")  where "⌊ψ⌋ ≡  ∀w. ψ w"
(*some tests: dualities*)
lemma "⌊(◇⪯φ)↔(¬□⪯¬φ)⌋ ∧ ⌊(◇≺φ)↔(¬□≺¬φ)⌋ ∧
       ⌊(Aφ)↔(¬E¬φ)⌋"    by blast (*proof*)
(**** Section 3: A basic modal preference language ****)
(*Definition 5*)
abbreviation p1::ν ("_⪯EE_")
             where "(φ ⪯EE ψ) u ≡ ∃s.∃t. φ s ∧ ψ t ∧ s⪯t"
abbreviation p2::ν ("_⪯AE_")
             where "(φ ⪯AE ψ) u ≡ ∀s.∃t. φ s ⟶ (ψ t ∧ s⪯t)"
abbreviation p3::ν ("_≺EE_")
             where "(φ ≺EE ψ) u ≡ ∃s.∃t. φ s ∧ ψ t ∧ s≺t"
abbreviation p4::ν ("_≺AE_")
             where "(φ ≺AE ψ) u ≡ ∀s.∃t. φ s ⟶ (ψ t ∧ s≺t)"
abbreviation p5::ν ("_≺AA_")
             where "(φ ≺AA ψ) u ≡ ∀s.∀t. (φ s ∧ ψ t) ⟶ s≺t"
abbreviation p6::ν ("_≻EA_")
             where "(φ ≻EA ψ) u ≡ ∃s.∀t. (φ s ∧ ψ t) ⟶ t≺s"
abbreviation p7::ν ("_⪯AA_")
             where "(φ ⪯AA ψ) u ≡ ∀s.∀t. (φ s ∧ ψ t) ⟶ s⪯t"
abbreviation p8::ν ("_≽EA_")
             where "(φ ≽EA ψ) u ≡ ∃s.∀t. (φ s ∧ ψ t) ⟶ t⪯s"
abbreviation P1::ν ("_⪯EE_") where "φ ⪯EE ψ ≡ E(φ ∧ ◇⪯ψ)"
abbreviation P2::ν ("_⪯AE_") where "φ ⪯AE ψ ≡ A(φ → ◇⪯ψ)"
abbreviation P3::ν ("_≺EE_") where "φ ≺EE ψ ≡ E(φ ∧ ◇≺ψ)"
abbreviation P4::ν ("_≺AE_") where "φ ≺AE ψ ≡ A(φ → ◇≺ψ)"
abbreviation P5::ν ("_≺AA_") where "φ ≺AA ψ ≡ A(ψ → □⪯¬φ)"
abbreviation P6::ν ("_≻EA_") where "φ ≻EA ψ ≡ E(φ ∧ □⪯¬ψ)"
abbreviation P7::ν ("_⪯AA_") where "φ ⪯AA ψ ≡ A(ψ → □≺¬φ)"
abbreviation P8::ν ("_≽EA_") where "φ ≽EA ψ ≡ E(φ ∧ □≺¬ψ)"
(*quantification for objects of arbitrary type*)
abbreviation mforall ("∀") where "∀Φ ≡ λw.∀x. (Φ x w)"
abbreviation mforallB (binder"∀"[8]9) where "∀x. φ(x) ≡ ∀φ"
abbreviation mexists ("∃") where "∃Φ ≡ λw.∃x. (Φ x w)"
abbreviation mexistsB (binder"∃"[8]9) where "∃x. φ(x) ≡ ∃φ"
(*polymorph operators for sets of worlds/values*)
abbreviation subs (infix "⊑" 70) where "A⊑B ≡ ∀x. A x ⟶ B x"
abbreviation union (infixr "⊔" 70) where "A⊔B ≡ λx. A x ∨ B x"
abbreviation inters (infixr "⊓" 70) where "A⊓B ≡ λx. A x ∧ B x"
(*Consistency confirmed (trivial: only abbreviations introduced)*)
lemma True nitpick[satisfy,user_axioms] oops
end
```

Figure 3: SSE of $\mathcal{PL}$ (Benthem et al. 2009) in HOL

SSE of $\mathcal{PL}$ in HOL We comment on the implementation of the SSE of $\mathcal{PL}$ in *Isabelle/HOL* as displayed in Fig. 3; see van Benthem et al. (2009) for further details on $\mathcal{PL}$.

First, a new base type i is declared (Line 7), denoting the set of possible worlds or states. Subsequently (Lines 8–12), useful type abbreviations are introduced, including the type σ for $\mathcal{PL}$ propositions, which are modeled as predicates on objects of type i (i.e. as *truth-sets* of worlds/states). A *betterness relation* $\preceq$, and its strict variant $\prec$, are introduced (Lines 14–15), 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 16–19).

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 30–31). 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.

As a first test some expected dualities of the modal operators are automatically proved (Lines 37–38).

Subsequently, the *betterness* ordering $\preceq$ (resp. $\prec$) 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 41–56). These are complemented by eight syntactic definitions of the same binary preferences stated within the object language $\mathcal{PL}$ (Lines 57–64).[3] Intuitively, preferring proposition P over Q amounts to preferring every P-state over every Q-state ($\prec_{AA}$), or at least one P-state over every Q-state ($\succ_{EA}$ or $\prec_{AE}$). Each of these non-trivial variants can be argued for, cf. (Benthem et al. 2009). However, only $\succ_{EA}$ and $\prec_{AE}$ satisfy the conditions required for a logic of value aggregation (see the discussion in §5). Transitivity, a quite controversial property in the literature on preferences, is satisfied by $\prec_{AE}$ but not by $\succ_{EA}$. Our framework can support both modeling options. In line with the *LogiKEy* methodology, such a choice is left at the discretion of the modeler.

We further extend the propositional logic $\mathcal{PL}$ as defined by van Benthem et al. (2009) by adding quantifiers (Lines 66–69).[4] Moreover, polymorphic operators for subset, union and intersection are defined (Lines 71–73).

The model finder *Nitpick* (Blanchette and Nipkow 2010) confirms the consistency of the introduced theory (Line 75) by generating and presenting a model for it (not shown here).

To gain practical evidence for the faithfulness of our SSE of $\mathcal{PL}$ in *Isabelle/HOL*, and also to assess proof automation performance, we have conducted numerous experiments in which we automatically reconstruct meta-theoretical results on $\mathcal{PL}$; see Figs. 10 and 11 in Appx. A.2.

5 Encoding the Value Ontology

The next, essential step is to model and encode the "Discoursive Grammar" value ontology. The current status of

[3] ATP systems can prove the meta-theoretic correspondences of these definitions; see Lines 5–13 in Fig. 10 in Appx. A.2.

[4] See Benzmüller and Paulson (2013) for a discussion of the SSE of quantified modal logics.

```
1  theory ValueOntology      (*** Benzmüller, Fuenmayor & Lomfeld, 2020 ***)
2    imports PreferenceLogicBasics
3  begin (*** Lomfeld's value ontology is encoded ***)
4
5  (*two legal parties (there can be more in principle)*)
6  datatype c = p | d (*parties/contenders: plaintiff, defendant*)
7  fun other::"c⇒c" ("_⁻¹") where "p⁻¹ = d" | "d⁻¹ = p"
8
9  consts For::"c⇒σ"    (*decision: find/rule for party*)
10 axiomatization where ForAx: "⌊For x ↔ (¬For x⁻¹)⌋"
11
12 datatype (*ethico-legal upper values (wrt. a given party)*)
13  't VAL = FREEDOM 't | UTILITY 't | SECURITY 't | EQUALITY 't
14 type_synonym v = "(c)VAL⇒bool" (*principles: sets of upper values*)
15 type_synonym cv = "c⇒v" (*principles are specified wrt. a given party*)
16
17 abbreviation vset1 ("{_}") where "{φ} ≡ λx::(c)VAL. x=φ"
18 abbreviation vset2 ("{_,_}")   where "{α,β} ≡ λx::(c)VAL. x=α ∨ x=β"
19
20 abbreviation utility::cv  ("UTILITY⁻")  where "UTILITYˣ  ≡ {UTILITY  x}"
21 abbreviation security::cv ("SECURITY⁻") where "SECURITYˣ ≡ {SECURITY x}"
22 abbreviation equality::cv ("EQUALITY⁻") where "EQUALITYˣ ≡ {EQUALITY x}"
23 abbreviation freedom::cv  ("FREEDOM⁻")  where "FREEDOMˣ  ≡ {FREEDOM  x}"
24 abbreviation stab::cv ("STAB⁻") where "STABˣ ≡ {SECURITY x, UTILITY  x}"
25 abbreviation effi::cv ("EFFI⁻") where "EFFIˣ ≡ {UTILITY  x, SECURITY x}"
26 abbreviation gain::cv ("GAIN⁻") where "GAINˣ ≡ {UTILITY  x, FREEDOM  x}"
27 abbreviation will::cv ("WILL⁻") where "WILLˣ ≡ {FREEDOM  x, UTILITY  x}"
28 abbreviation resp::cv ("RESP⁻") where "RESPˣ ≡ {FREEDOM  x, EQUALITY x}"
29 abbreviation fair::cv ("FAIR⁻") where "FAIRˣ ≡ {EQUALITY x, FREEDOM  x}"
30 abbreviation equi::cv ("EQUI⁻") where "EQUIˣ ≡ {EQUALITY x, SECURITY x}"
31 abbreviation reli::cv ("RELI⁻") where "RELIˣ ≡ {SECURITY x, EQUALITY x}"
32
33 (*derivation operators (cf. theory of "formal concept analysis") *)
34 consts Vrel::"i⇒(c)VAL⇒bool" ("ℐ") (*incidence relation worlds-values*)
35 abbreviation intension::"σ⇒v" ("_↑") where "W↑ ≡ λv. ∀x. W x ⟶ ℐ x v"
36 abbreviation extension::"v⇒σ" ("_↓") where "V↓ ≡ λw. ∀x. V x ⟶ ℐ w x"
37
38 (*shorthand notation for aggregating values*)
39 abbreviation agg (infixr "⊕"80) where "v₁⊕v₂ ≡ v₁ ⊓ v₂"
40 abbreviation agg1 ("[_]") where "[v] ≡ v↓"
41 abbreviation agg2 ("[_⊕_]") where "[v₁⊕v₂] ≡ (v₁⊕v₂)↓"
42 abbreviation agg3 ("[_⊕_⊕_]") where "[v₁⊕v₂⊕v₃] ≡ (v₁⊕v₂⊕v₃)↓"
43 abbreviation agg4 ("[_⊕_⊕_⊕_]") where "[v₁⊕v₂⊕v₃⊕v₄] ≡ (v₁⊕v₂⊕v₃⊕v₄)↓"
44
45 (*chosen variant for preference relation (cf. van Benthem et al. 2009*)
46 abbreviation relPref::"σ⇒σ⇒σ"  ("_≺_") where "φ ≺ ψ ≡ ψ ≻EA φ"
47 abbreviation relPrefval::"v⇒v⇒σ" ("_≺ᵥ_") where "φ ≺ᵥ ψ ≡ ψ↓ ≻EA φ↓"
48
49 abbreviation incnst ("INCONS⁻") where (*inconsistency for value support*)
50  "INCONSˣ ≡ [SECURITYˣ] ∧ [EQUALITYˣ] ∧ [FREEDOMˣ] ∧ [UTILITYˣ]"
51
52 lemma "True" nitpick[satisfy] oops (*verify consistency of this theory*)
53 end
```

Figure 4: Encoding of the Value Ontology

our modeling efforts is illustrated in the *Isabelle/HOL* theory displayed in Fig. 4. Similar to before, this modeling may be subject to further modifications. In fact, what is presented here is the result of several cycles of modeling, encoding and testing activities, in which previous versions have been refuted and rejected or further improved.

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 6–7); and a predicate modeling the ruling *for* a party is also provided (Lines 9–10).

As regards to Lomfeld's upper value ontology, a four-valued (parameterized) datatype is introduced (Lines 12–13) as described in §2. Moreover, type-aliases (Lines 14–15) and set-constructor operators for values (Lines 17–18) are introduced for ease of presentation. The legal principles from §2 are next introduced as combinations of those upper values (Lines 20–31). As an illustration, the principle *stability* (STAB) is understood as instantiation of the upper value

```
1  theory ValueOntologyTest (*** Benzmüller, Fuenmayor & Lomfeld, 2020 ***)
2    imports ValueOntology
3  begin (* value ontology tests *)
4  (*values in two opposed quadrants: inconsistent*)
5  lemma "⌊[RESPˣ] ∧ [STABˣ] ⟶ INCONSˣ⌋" by simp
6  lemma "⌊[RELIˣ] ∧ [WILLˣ] ⟶ INCONSˣ⌋" by simp
7  (*all values in two non-opposed quadrants: consistent*)
8  lemma "⌊[WILLˣ] ∧ [STABˣ] ⟶ INCONSˣ⌋" nitpick oops  (*countermodel*)
9  (*values in opposed quadrants for different parties: consistent*)
10 lemma "⌊[EQUIˣ] ∧ [GAINʸ] ⟶ (INCONSˣ ∨ INCONSʸ)⌋" nitpick oops (*ctm*)
11 lemma "⌊[RESPˣ] ∧ [STABʸ] ⟶ (INCONSˣ ∨ INCONSʸ)⌋" nitpick oops (*ctm*)
12 lemma "⌊[RELIᵖ] ∧ [WILLᵖ]⌋" nitpick[satisfy] nitpick oops (*contingent*)
13 (*value preferences tests*)
14 lemma "⌊WILLˣ ≺ᵥ STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ RELIˣ⊕STABˣ⌋" by blast
15 lemma "⌊RELIˣ⊕STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊STABˣ ≺ᵥ WILLˣ⌋" by auto
16 lemma "⌊WILLˣ ≺ᵥ RELIˣ⊕STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ STABˣ⌋"
17   nitpick nitpick[satisfy] oops (*contingent*)
18 lemma "⌊STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊RELIˣ⊕STABˣ ≺ᵥ WILLˣ⌋"
19   nitpick nitpick[satisfy] oops (*contingent*)
20 end
```

Figure 5: Testing the Value Ontology

security coalesced with *utility* (in that order of importance).[5]

After defining legal principles as combinations (in this case: sets) of values (w.r.t. a legal party), we need to relate them to propositions (sets of worlds/states) in our logic $\mathcal{PL}$. For this we reuse some basic notions from the theory of *Formal Concept Analysis* (FCA) (Ganter and Wille 2012). We define a binary *incidence relation* (Line 34) between worlds/states (type i) and upper values (type *VAL*), which is used to link propositions (sets of worlds) with corresponding sets of values (their *intension*) by means of the operator $\uparrow$ (Line 35). More important to our purposes: each legal principle (set of values) is associated with a proposition (its *extension*) by means of the operator $\downarrow$ (Line 36).[6]

Intuitively, we can read the proposition denoted by $\text{STAB}^p\downarrow$ as: "a decision in favor of the plaintiff (p) promotes the principle of stability". Value-inconsistency (INCONS) is given when all four upper values apply for the same legal party (Lines 49–50). Further shorthand notation for value aggregation is introduced (Lines 39–43).

The preference relation $\prec$ utilized in the remainder is chosen to be the $\succ_{EA}$-variant from §4 (Line 46–47). As mentioned before, the (better-known, transitive) $\prec_{AE}$-variant is also suitable for our logic of value preferences. The results presented are valid for both (unless otherwise stated).

Testing the Value Ontology

In order to test the adequacy of our modeling, some implied and non-implied knowledge is studied. This is shown in Fig. 5. We briefly discuss some of the conducted tests.

Following a conflictual dialectical interpretation of the value ontology, promoting values (for the same party) from two opposing quadrants, say RESP & STAB or RELI & WILL (cf. Fig. 1), is provably 'value-inconsistent' (INCONS); theorem provers quickly confirm this (Lines 5-6). However, promoting values from two non-opposed quad-

[5] At this early stage of our modeling we have not yet considered this additional factor of relative importance or weight between values. Doing this would allow us to clearly distinguish, e.g., *stability* (STAB) from *efficiency* (EFFI) among others.

[6] FCA exhibits a perfect duality between the pairs of notions: *object/attribute* and *extension/intension*. Our choice to consider worlds as *objects* and values as their *attributes* (i.e. propositions as *extensions* of legal principles) bears no mathematical significance.

```
Nitpick found a model for card i = 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. _)((i₁, i₁) := True)
    For = (λx. _)((d, i₁) := False, (p, i₁) := True)
    ℐ = (λx. _)
        ((i₁, FREEDOM d) := False,
         (i₁, FREEDOM p) := True,
         (i₁, UTILITY d) := False,
         (i₁, UTILITY p) := True,
         (i₁, EQUALITY d) := False,
         (i₁, EQUALITY p) := True,
         (i₁, SECURITY d) := False,
         (i₁, SECURITY p) := True)
    other = (λx. _)(d := p, p := d)
```

Figure 6: Satisfying model for the statement in Line 12 of Fig. 5.

rants, such as WILL & STAB (Line 8) is value-consistent: the model finder *Nitpick* computes and reports a countermodel (not shown here) to the stated conjecture. Value-inconsistency is also not implied if values from opposing quadrants are promoted for different parties (Lines 10–11).

Remark on model finder *Nitpick* (Blanchette and Nipkow 2010): *Nitpick* searches for, 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'. In Line 12 of Fig. 5, for example, *Nitpick* is called simultaneously in both modes in order to confirm the contingency of the statement; as expected both a model (Fig. 6) and countermodel for the statement is returned. These models are given as concrete interpretations of relevant terms in the given context so that the conjectured statement is satisfied or falsified (depending of the mode of *Nitpick*). Studying and analysing the returned interpretations can be very helpful and intuition-fostering for the user, in particular, in very complex modeling tasks when intuition and deep understanding of the modeled structures is initially lacking.

Note that the notion of value-inconsistency (INCONS), cf. Lines 49–50 in Fig. 4, has deliberately not been aligned with inconsistency in meta-logic HOL so that it can be explicitly reasoned with (avoiding explosion). That means, that we can contingently promote conflicting values such as RELI and WILL for p, and this then leads to value-inconsistency (for p) but not to general inconsistency in meta-logic HOL. In Line 12 this contingency is confirmed. However, inspecting the satisfying models generated by *Nitpick*, we see that p is indeed value-inconsistent. One of such models is depicted in Fig. 6, where it is shown that (in the given possible world i_1) the upper values EQUALITY, SECURITY, UTILITY and FREEDOM are simultaneously promoted for p, this amounts to value-inconsistency (INCONS) according to our definition.

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 case of issues then triggers adaptions of the defining axioms. Such a process may require several cycles (remember the discussion from §3) and, as a useful side effect, it clearly fosters cross-disciplinary mutual understanding.

Further tests in Fig. 5 assess the suitability of the preference relation $\prec_v$ (either $\prec_{AE}$ or $\succ_{EA}$) for reasoning e.g. with value aggregations. For example, we test for a correct behavior 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 14). Similar test are conducted for 'weakening' of the left-hand side.

6 General Legal & World Knowledge

The realistic modeling of concrete legal cases requires further legal & world knowledge (LWK) to be taken into account. LWK is typically modeled 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 present in Fig. 7 only a small and monolithic example theory called "GeneralKnowledge" for illustration. In our case this includes a small excerpt of a much simplified "animal appropriation taxonomy". Moreover, we associate "animal appropriation" (kinds of) situations with the value preferences they imply (conditional preference relations).

In a realistic setting this knowledge base would be further split and structured similarly to other legal or general ontologies, e.g., in the *Semantic Web*. 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 modeling, 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.

Next steps include interrelating certain notions in our theory "GeneralKnowledge" with values, resp. value preferences, as introduced in the previous sections. It is here where the preference relations and modal operators of $\mathcal{PL}$ are used. 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 a preference logic, or we may replace our current material implication operator by a conditional implication to better support defeasible legal reasoning.[7]

A general problem of knowledge representation frameworks and ontology languages often is their insufficient support for the proper treatment of modalities, see e.g. the discussion and solution proposed by Benzmüller and Pease (2012). Our framework, which is based on that proposed solution, is not suffering from such restrictions.

We briefly explain below the *Isabelle/HOL* encoding of our LWK as shown in Fig. 7.

First, some non-logical constants that stand for kinds of legally relevant situations (here: of appropriation) are introduced (Lines 6–9), and their meaning is constrained by some postulates (Lines 12–17). As already discussed, for ease of illustration, these terms are modeled here as simple propositional constants. In practice, however, they may later be replaced, or logically implied, by a more realistic and suitable

[7] Indeed a (cautiously-monotonic) conditional implication can be defined employing $\mathcal{PL}$ modal operators. We may also opt for an SSE of a conditional logic in HOL, see e.g. Benzmüller (2013).

```
theory GeneralKnowledge (*Benzmüller, Fuenmayor & Lomfeld, 2020*)
  imports ValueOntology
begin (*** General Legal and World Knowledge (LWK) ***)

(*LWK: kinds of situations addressed*)
consts appObject::"σ"      (*appropriation of objects in general*)
consts appAnimal::"σ"      (*appropriation of animals in general*)
consts appWildAnimal::"σ"  (*appropriation of wild animals*)
consts appDomAnimal::"σ"   (*appropriation of domestic animals*)

(*LWK: postulates for kinds of situations*)
axiomatization where
 W1: "⌊(appWildAnimal ∨ appDomAnimal) ↔ appAnimal⌋" and
 W2: "⌊appWildAnimal ↔ ¬appDomAnimal⌋" and
 W3: "⌊appWildAnimal → appAnimal⌋"  and
 W4: "⌊appDomAnimal → appAnimal⌋" and
 W5: "⌊appAnimal → appObject⌋"
(*...further situations regarding appropriation of objects, etc.*)

(*LWK: (prima facie) value preferences for kinds of situations*)
axiomatization where
 R1: "⌊appAnimal → (STAB^p ≺_v STAB^d)⌋" and
 R2: "⌊appWildAnimal → (WILL^x⁻¹ ≺_v STAB^x)⌋" and
 R3: "⌊appDomAnimal  → (STAB^x⁻¹ ≺_v RELI^x⊕RESP^x)⌋"
(*...further preferences...*)

(*LWK: domain vocabulary*)
typedecl e    (*declares new type for 'entities'*)
consts Animal::"e⇒σ"
consts Domestic::"e⇒σ"
consts Fox::"e⇒σ"
consts Parrot::"e⇒σ"
consts Pet::"e⇒σ"
consts FreeRoaming::"e⇒σ"

(*LWK: taxonomic (domain) knowledge*)
axiomatization where
 W6: "⌊∀a. Fox a → Animal a⌋" and
 W7: "⌊∀a. Parrot a → Animal a⌋" and
 W8: "⌊∀a. (Animal a ∧ FreeRoaming a ∧ ¬Pet a) → ¬Domestic a⌋"
(*...others...*)

(*LWK: legally-relevant, situational 'factors'*)
consts Own::"c⇒σ"    (*object is owned by party c*)
consts Poss::"c⇒σ"   (*party c has actual possession of object*)
consts Intent::"c⇒σ" (*party c has intention to possess object*)
consts Mal::"c⇒σ" (*party c acts out of malice*)
consts 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)⌋"
 (*...others...*)

(*LWK: conditional value preferences, e.g. from precedents*)
axiomatization where
 R4: "⌊(Mal x⁻¹ ∧ Own x)  → (STAB^x⁻¹ ≺_v RESP^x⊕RELI^x)⌋"
 (*...others...*)

(*LWK: relate values, outcomes and situational 'factors'*)
axiomatization where
 F1: "⌊For x → (Intent x ↔ □≤[WILL^x])⌋" and
 F2: "⌊For x → (Mal x⁻¹ ↔ □≤[RESP^x])⌋" and
 F3: "⌊For x → (Poss x ↔ □≤[STAB^x])⌋" and
 F4: "⌊For x → (Mtn x ↔ □≤[RESP^x])⌋" and
 F5: "⌊For x → (Own x ↔ □≤[RELI^x])⌋"

(*theory is consistent, (non-trivial) model found*)
lemma True nitpick[satisfy,card i=10] oops
end
```

Figure 7: Encoding of general legal & world knowledge

modeling utilizing arbitrarily complex (even higher-order, if needed) formulas depicting states of affairs to some desired level of granularity. Some simple vocabulary and taxonomic relationships (here: for wild and domestic animals) are specified (Lines 28–40) to illustrate this.

The legal default rules for several situations (here: appropriation of animals) are formulated as *prima facie* preference relations (Lines 21–24). For example, one rule R2 (Line 23) could be read as: "In a wild-animals-appropriation kind of situation, a decision promoting STAB in favor of a party (say, the plaintiff) is preferred over a decision, favoring the other party (defendant), which promotes WILL". If there is no more specific legal rule from a precedent or a codified statute then these *prima facie* preference relations determine the result. As a didactic example, the legal rule R4 (Line 58) states that the ownership (say, the plaintiff's) of the land on which the hunting took place, together with the fact that the opposing party (defendant) acted out of malice implies a value preference of *reliance* and *responsibility* over *stability*, cf. §2. This reflects the Common law precedent of Keeble v. Hickergill (1704, 103 ER 1127).

An established AI & Law tool to structure the representation of legal precedents are situational "factors" (Ashley 1990; Prakken and Sartor 2015). Some of them are specified as illustration (Lines 44–48), together with some postulates constraining their meaning (Lines 51–53). Our framework also allows us to introduce definitions for those factors for which clear legal specifications exist. In our model, such factors are further related to value principles and outcomes (Lines 62–67). Our normative assignment here is widely in accordance with classifications in the AI & Law literature (Berman and Hafner 1993; Bench-Capon 2012).

Finally, the consistency of all axioms and rules provided is confirmed by *Nitpick* (Line 70).

7 Proof of Concept – Case Study

We illustrate our reasoning framework by encoding the classic common law property case Pierson v. Post. In a nutshell:

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

The modeling is presented in Fig. 8. We have introduced and interrelated some minimal vocabulary, e.g. to "pursue" and to "capture" (Lines 6–14), needed to represent the case facts as interpreted by the parties (Lines 19–20 for Pierson; Lines 45–46 for Post). Their consistency with other postulates from the previously introduced layers "GeneralKnowledge" and "ValueOntology" (imported in Line 2) is verified by generating a (non-trivial) model using *Nitpick* (Line 16).

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 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* (STAB) and outweighs the mere *will* to possess (WILL) 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 legal value ontology in §2, this corresponds to a preference for the (upper) value SECURITY over FREEDOM. We can see that this legal rule R2,

as previously introduced in the layer "GeneralKnowledge" (see Fig. 7, Line 23) is indeed employed by *Isabelle/HOL* automated tools to prove (Lines 31–32) that the given facts imply a preference for a decision for Pierson over one for Post. Notice that the legal precedent rule R4 of Keeble v. Hickergill (see Fig. 7, Line 58) does not apply to this case.

We also present and model a possible counterargument for 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" as such (Line 45–46). Moreover, an alternative legal rule (i.e. a possible argument for overruling in case law methodology) is presented (Lines 42–43), entailing a value preference of *efficiency* (EFFI) over *stability* (STAB), and justified by the alleged public benefit of hunters getting rid of foxes, since the latter cause depredations in farms. This is the argument put forward by the dissenting opinion in the original case (3 Cai R 175). Other tests, i.e. consistency of both decision alternatives with the given premises and with the value-consistency of both parties (Lines 48–54) and refutability of a possible decision favoring Pierson (Lines 61–62), are analogous (but contrary) to Pierson's argument.

8 Conclusion

Contributions and preliminary results of an ongoing project have been presented in which a "discursive grammar" value ontology in combination with further legal and general world knowledge is being utilized for a detailed, granular assessment and explanation of legal argumentation as partly controversial legal balancing in a concrete case.

From a technical perspective, the core objective of this paper has been to demonstrate that the technology we promote and apply—shallow semantical embeddings (SSEs) in classical higher-order logic (HOL)—appears indeed suitable for the task of structured legal balancing. It is the flexibility of the multi-layer modeling which is unique in our approach, in combination with a very rich support for expressiveness, quantified classical and non-classical logics, thereby rejecting the idea that knowledge representation means should or must be limited a priori to propositional frameworks due only to computational considerations.

From a legal perspective, the reconstruction of legal balancing is already with classical argumentative tools a non-trivial task which is methodologically not settled at all (Sieckmann 2010). Here, our paper proposed the structuring of legal argumentation with a dialectical ethico-legal value ontology. Legal rules and their various interpretations could thus be displayed within a unified pluralistic logic of value preference relations. The integration of the preference logic and the value ontology within the dynamic HOL modeling environment allows to experiment with different forms of interpretation. This enables not only to find more accurate reconstructions of legal argumentation but also supports to aggregate legal arguments and check their normative (value) consistency. We aim at expanding the model towards proportional preference relations which would allow to integrate qualitative and quantitative legal balancing approaches and to evaluate their relative strength.

```
1  theory Pierson    (*** Benzmüller, Fuenmayor & Lomfeld, 2020 ***)
2    imports GeneralKnowledge
3  begin (*** Pierson v. Post "wild animal" case **)
4
5  (*case-specific 'world-vocabulary'*)
6  consts α::"e" (*appropriated animal (fox in this case) *)
7  consts Pursue::"c⇒e⇒σ"
8  consts Capture::"c⇒e⇒σ"
9
10 (*case-specific taxonomic (legal domain) knowledge*)
11 axiomatization where
12  CW1: "⌊(∃c. Capture c α ∧ ¬Domestic α) → appWildAnimal⌋" and
13  CW2: "⌊∀c. Pursue c α → Intent c⌋" and
14  CW3: "⌊∀c. Capture c α → Poss c⌋"
15
16 lemma True nitpick[satisfy,card i=4] oops (*satisfiable*)
17
18 (***************** pro-Pierson's argument ***************)
19 abbreviation "Pierson_facts ≡ ⌊Fox α ∧ (FreeRoaming α) ∧
20   (¬Pet α) ∧ Pursue p α ∧ (¬Pursue d α) ∧ Capture d α⌋"
21
22 (*decision for defendant (Pierson) is compatible with premises*)
23 lemma "Pierson_facts ∧ ⌊¬INCONS^P⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊For p ≺ For d⌋"
24   nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)
25
26 (*decision for plaintiff (Post) is compatible with premises*)
27 lemma "Pierson_facts ∧ ⌊¬INCONS^P⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊For d ≺ For p⌋"
28   nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)
29
30 (*decision for defendant (Pierson) is provable*)
31 theorem assumes Pierson_facts shows "⌊For p ≺ For d⌋"
32   by (metis assms CW1 CW2 W6 W8 ForAx R2 F1 other.simps(2) rBR)
33
34 (*while a decision for the plaintiff is not*)
35 lemma assumes Pierson_facts shows "⌊For d ≺ For p⌋"
36   nitpick[card i=4] oops (*counterexample found*)
37
38 (***************** pro-Post's argument ***************)
39 (* Theory amendment: the animal is not free-roaming since it
40    is being chased by a professional hunter (Post) *)
41 consts Hunter::"c⇒σ"
42 axiomatization where (*case-specific legal rule for hunters*)
43  R5: "⌊(Hunter x ∧ Pursue x α)  → (STAB^(x-1) ≺_v EFFI^x)⌋"
44
45 abbreviation "Post_facts ≡ ⌊Fox α ∧ (¬FreeRoaming α) ∧
46    Hunter p ∧ Pursue p α ∧ (¬Pursue d α) ∧ Capture d α⌋"
47
48 (*decision for defendant (Pierson) is compatible with premises*)
49 lemma "Post_facts ∧ ⌊¬INCONS^P⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊For p ≺ For d⌋"
50   nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)
51
52 (*decision for plaintiff (Post) is compatible with premises too*)
53 lemma "Post_facts ∧ ⌊¬INCONS^P⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊For d ≺ For p⌋"
54   nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)
55
56 (*indeed, a decision for plaintiff (Post) now becomes provable*)
57 theorem assumes Post_facts shows "⌊For d ≺ For p⌋"
58   using assms by (metis CW3 ForAx R5 F3 other.simps rBR)
59
60 (*while a decision for the defendant is now refutable*)
61 lemma assumes Post_facts shows "⌊For p ≺ For d⌋"
62   nitpick[card i=4] oops (* counterexample found*)
63 end
```

Figure 8: Modeling the Pierson v. Post case.

The suitable, reusable, paradox-free modeling of ethico-legal balancing as presented here is partly an art and partly a challenging engineering task. Combining both aspects in a holistic approach within a dynamic modeling framework, we hope to contribute to the evolution of legal reasoning and to pave the way for some form of (legally) reasonable machines.

Acknowledgements: Will be included in the final document.

References

Alexy, R., ed. (1978). *Theorie der juristischen Argumentation*. Frankfurt/M: Suhrkamp.

– (2000). "On the Structure of Legal Principles". In: *Ratio Juris* 13, pp. 294–304.

– (2003). "On Balancing and Subsumption: A Structural Comparison". In: *Ratio Juris* 16, pp. 433–449.

Ashley, K. D. (1990). *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge/MA: MIT Press.

Bench-Capon, T. (2012). "Representing Popov v Hayashi with dimensions and factors". In: *Artificial Intelligence and Law* 20, pp. 15–35.

Bench-Capon, T., K. Atkinson, and A. Chorley (2005). "Persuasion and value in legal argument". In: *Journal of Logic and Computation* 15, pp. 1075–1097.

Bench-Capon, T. and G. Sartor (2003). "A model of legal reasoning with cases incorporating theories and value". In: *Artificial Intelligence* 150, pp. 97–143.

Benthem, J. van, P. Girard, and O. Roy (2009). "Everything Else Being Equal: A Modal Logic for *Ceteris Paribus* Preferences". In: *J. Philos. Log.* 38.1, pp. 83–125.

Benzmüller, C. (2013). "Automating Quantified Conditional Logics in HOL". In: *IJCAI-13*. Ed. by F. Rossi. AAAI Press, pp. 746–753.

– (2019). "Universal (Meta-)Logical Reasoning: Recent Successes". In: *Sci. Comp. Progr.* 172, pp. 48–62.

Benzmüller, C. and B. Lomfeld (2020). "Reasonable Machines: A Research Manifesto". Preprint: https://dx.doi.org/10.13140/RG.2.2.28918.63045.

Benzmüller, C., X. Parent, and L. van der Torre (2020). "Designing Normative Theories for Ethical and Legal Reasoning: LogiKEy Framework, Methodology, and Tool Support". In: *Artificial Intelligence*. Doi: https://doi.org/10.1016/j.artint.2020.103348.

Benzmüller, C. and L. C. Paulson (2013). "Quantified Multimodal Logics in Simple Type Theory". In: *Logica Universalis* 7.1, pp. 7–20.

Benzmüller, C. and A. Pease (2012). "Higher-order Aspects and Context in SUMO". In: *Journal of Web Semantics* 12-13, pp. 104–117.

Berman, D. and C. Hafner (1993). "Representing teleological structure in case-based legal reasoning: the missing link". In: *Proceedings 4th ICAIL*. New York: ACM Press, pp. 50–59.

Blanchette, J. C. and T. Nipkow (2010). "Nitpick: A Counterexample Generator for Higher-Order Logic Based on a Relational Model Finder". In: *ITP 2010*. Vol. 6172. LNCS. Springer, pp. 131–146.

Dung, P. M. (1995). "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games". In: *Artificial Intelligence* 77, pp. 321–357.

Fuenmayor, D. and C. Benzmüller (2019a). "A Computational-Hermeneutic Approach for Conceptual Explicitation". In: *Model-Based Reasoning in Science and Technology. Inferential Models for Logic, Language, Cognition and Computation*. Vol. 49. SAPERE. Springer, Cham, pp. 441–469.

– (2019b). "Harnessing Higher-Order (Meta-)Logic to Represent and Reason with Complex Ethical Theories". In: *PRICAI 2019: Trends in Artificial Intelligence*. Vol. 11670. LNAI. Springer, Cham, pp. 418–432.

Ganter, B. and R. Wille (2012). *Formal concept analysis: mathematical foundations*. Springer Berlin.

Gordon, T. and D. Walton (2012). "A Carneades reconstruction of Popov v Hayashi". In: *Artificial Intelligence and Law* 20, pp. 37–56.

Hage, J. (1997). *Reasoning With Rules*. Dordrecht: Kluwer.

Hoekstra, R., J. Breuker, M. D. Bello, and A. Boer (2009). "LKIF Core: Principled Ontology Development for the Legal Domain". In: *Law, Ontologies and the Semantic Web - Channelling the Legal Information Flood*. Vol. 188. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 21–52.

Horty, J. (2011). "Rules and reasons in the theory of precedent". In: *Legal Theory* 17, pp. 1–33.

Liu, F. (2008). "Changing for the Better – Preference Dynamics and Agent Diversity". PhD thesis. Inst. for Logic, Language and Computation, Universiteit van Amsterdam.

Lomfeld, B. (2015). *Die Gründe des Vertrages: Eine Diskurstheorie der Vertragsrechte*. Tübingen: Mohr Siebeck.

– (2017). "Vor den Fällen: Methoden soziologischer Jurisprudenz". In: *Die Fälle der Gesellschaft: Eine neue Praxis soziologischer Jurisprudenz*. Ed. by Lomfeld. Tübingen: Mohr Siebeck, pp. 1–16.

– (2019). "Grammatik der Rechtfertigung: Eine kritische Rekonstruktion der Rechts(fort)bildung". In: *Kritische Justiz* 52.4.

Nipkow, T., L. Paulson, and M. Wenzel (2002). *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Vol. 2283. Lecture Notes in Computer Science. Springer.

Osherson, D. N. and S. Weinstein (2012). "Preference based on Reasons". In: *Rev. Symb. Log.* 5.1, pp. 122–147.

Prakken, H. and G. Sartor (2015). "Law and logic: A review from an argumentation perspective". In: *Artificial Intelligence* 227, pp. 214–225.

Sartor, G. (2010). "Doing justice to rights and values: teleological reasoning and proportionality". In: *Artificial Intelligence and Law* 18, pp. 175–215.

– (2018). "A Quantitative Approach to Proportionality". In: *Handbook of Legal Reasoning and Argumentation*. Ed. by B. et al. Dordrecht: Springer, pp. 613–636.

Sieckmann, J.-R., ed. (2010). *Legal Reasoning: The Methods of Balancing*. Vol. 124. ARSP Beiheft. Stuttgart: Franz Steiner.

Teubner, G. (1983). "Substantive and Reflexive Elements in Modern Law". In: *Law & Society Rev.* 17, pp. 239–285.

A Appendix

A.1 Ceteris paribus preference relation for $\mathcal{PL}$

Extending the SSE of $\mathcal{PL}$ in HOL from Fig. 3 some further preference relations for $\mathcal{PL}$ are defined in Fig. 9. These additional relations support *ceteris paribus* reasoning in $\mathcal{PL}$.

```
theory PreferenceLogicCeterisParibus (** Benzmüller & Fuenmayor, 2020 **)
  imports PreferenceLogicBasics
begin (** Ceteris Paribus reasoning by van Benthem et al, JPL 2009 **)

(*Section 5: Equality-based Ceteris Paribus Preference Logic*)
abbreviation a1::"σ⇒π⇒bool" ("_∈_") where "φ ∈ Γ ≡ Γ φ"
abbreviation a2 ("_⊆_") where "Γ ⊆ Γ' ≡ ∀φ. φ ∈ Γ ⟶ φ ∈ Γ'"
abbreviation a3 ("_∪_") where "Γ ∪ Γ' ≡ λφ. φ ∈ Γ ∨ φ ∈ Γ'"
abbreviation a4 ("_∩_") where "Γ ∩ Γ' ≡ λφ. φ ∈ Γ ∧ φ ∈ Γ'"
abbreviation a5 ("{_}") where "{φ} ≡ λx::σ. x=φ"
abbreviation a6 ("{_,_}")   where "{α,β} ≡ λx::σ. x=α ∨ x=β"
abbreviation a7 ("{_,_,_}") where "{α,β,γ} ≡ λx::σ. x=α ∨ x=β ∨ x=γ"
abbreviation a8 ("∅")   where "∅ ≡ (λψ::σ. False)"
abbreviation a9 ("𝒰")   where "𝒰 ≡ (λψ::σ. True)"

abbreviation c14 ("_≡_ _") where "w ≡Γ v ≡ ∀φ. φ ∈ Γ ⟶ (φ w ⟷ φ v)"
abbreviation c15 ("_⊴_") where "w ⊴Γ v ≡ w ≼ v ∧ w ≡Γ v"
abbreviation c16 ("_◁_") where "w ◁Γ v ≡ w ≺ v ∧ w ≡Γ v"
abbreviation c17 ("⟨_⟩⁼_") where "⟨Γ⟩⁼φ ≡ λw.∃v. w ⊴Γ v ∧ φ v"
abbreviation c18 ("[_]⁼_") where "[Γ]⁼φ ≡ λw.∀v. w ⊴Γ v ⟶ φ v"
abbreviation c19 ("⟨_⟩˂_") where "⟨Γ⟩˂φ ≡ λw.∃v. w ◁Γ v ∧ φ v"
abbreviation c20 ("[_]˂_") where "[Γ]˂φ ≡ λw.∀v. w ◁Γ v ⟶ φ v"
abbreviation c21 ("⟨_⟩_")  where "⟨Γ⟩φ ≡ λw.∃v. w ≡Γ v ∧ φ v"
abbreviation c22 ("[_]_")  where "[Γ]φ ≡ λw.∀v. w ≡Γ v ⟶ φ v"

(*Section 6: Ceteris Paribus Counterparts of Binary Pref. Statements*)
(*operators below not defined in paper; existence is tacitly suggested.
  AA-variant draws upon von Wright's. AE-variant draws upon Halpern's.*)
abbreviation C23 ("_≺AA_")
  where "(φ ≺AA^Γ ψ) u ≡ ∀s.∀t. φ s ∧ ψ t ⟶ s ◁Γ t"
abbreviation c24 ("_≼AA_")
  where "(φ ≼AA^Γ ψ) u ≡ ∀s.∀t. φ s ∧ ψ t ⟶ s ⊴Γ t"
abbreviation c25 ("_≺AE_")
  where "(φ ≺AE^Γ ψ) u ≡ ∀s.∃t. φ s ⟶ ψ t ∧ s ◁Γ t"
abbreviation c26 ("_≼AE_")
  where "(φ ≼AE^Γ ψ) u ≡ ∀s.∃t. φ s ⟶ ψ t ∧ s ⊴Γ t"
abbreviation c27 ("_≺AA_") where "φ ≺AA^Γ ψ ≡ A(ψ → [Γ]⁼¬φ)"
abbreviation c28 ("_≼AA_") where "φ ≼AA^Γ ψ ≡ A(ψ → [Γ]˂¬φ)"
abbreviation c29 ("_≺AE_") where "φ ≺AE^Γ ψ ≡ A(φ → ⟨Γ⟩˂ψ)"
abbreviation c30 ("_≼AE_") where "φ ≼AE^Γ ψ ≡ A(φ → ⟨Γ⟩⁼ψ)"

(*Consistency confirmed (trivial: only abbreviations are introduced*)
lemma True nitpick[satisfy,user_axioms] oops
end
```

Figure 9: SSE (cont'd) of $\mathcal{PL}$ (Benthem et al. 2009) in HOL

We give some explanations:

Lines 6–14 Useful set theoretic notions are introduced as abbreviations for corresponding λ-terms in HOL.

Lines 16–24 $\mathcal{PL}$ 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 modeling von Wright's notion of *ceteris paribus* ("all other things being equal") preferences, eliciting an appropriate Γ by extra-logical means.

Lines 29–40: Except for $\prec^{\Gamma}_{AA}$, the remaining operators we define here are not explicitly defined in (Benthem et al. 2009); however, their existence is tacitly suggested.

A.2 Testing the Meta-Theory of $\mathcal{PL}$

Meta-theoretical results on $\mathcal{PL}$ as presented by van Benthem et al. (2009) are automatically verified by the reasoning tools in *Isabelle/HOL*; see Figs. 10 and 11.

```
theory PreferenceLogicTests1      (*** Benzmüller & Fuenmayor, 2020 ***)
  imports PreferenceLogicBasics
begin (*Tests for the SSE of van Benthem et al, JPL 2009, in HOL*)
(*Fact 1: definability of the principal operators and verification*)
lemma F1_9:  "(φ ≼EE ψ) u ⟷ (φ ≼EE ψ) u" by smt
lemma F1_10: "(φ ≼AE ψ) u ⟷ (φ ≼AE ψ) u" by smt
lemma F1_11: "(φ ≺EE ψ) u ⟷ (φ ≺EE ψ) u" by smt
lemma F1_12: "(φ ≺AE ψ) u ⟷ (φ ≺AE ψ) u" by smt
(*Fact 2: definability of remaining pref. operators and verification*)
lemma F2_13: "is_total SBR ⟶ ((φ ≺AA ψ) u ⟷ (φ ≺AA ψ) u)"  by smt
lemma F2_14: "is_total SBR ⟶ ((φ ≻EA ψ) u ⟷ (φ ≻EA ψ) u)"  by smt
lemma F2_15: "is_total SBR ⟶ ((φ ≼AA ψ) u ⟷ (φ ≼AA ψ) u)"  by smt
lemma F2_16: "is_total SBR ⟶ ((φ ≽EA ψ) u ⟷ (φ ≽EA ψ) u)"  by smt
(*Section 3.5 "Axiomatization" -- verify interaction axioms*)
lemma Incl_1:   "⌊(◇˂φ) → (◇⁼φ)⌋"       by auto
lemma Inter_1:  "⌊(◇⁼◇˂φ) → (◇˂φ)⌋"    using tBR by blast
lemma Trans_le: "⌊(◇˂◇˂φ) → (◇˂φ)⌋"    using tBR by blast
lemma Inter_2:  "⌊(φ ∧ ◇⁼ψ) →  ((◇˂ψ) ∨ ◇⁼(ψ ∧  ◇⁼φ))⌋" by blast
lemma F4: "⌊(φ ∧ ◇⁼ψ) → ((◇˂ψ) ∨ ◇⁼(ψ ∧  ◇⁼φ))⌋ ⟷
           (∀w. ∀v. (((w ≼ v) ∧ ¬(v ≼ w)) ⟶ (w ≺ v)))" by smt
lemma Inter_3: "⌊(◇⁼◇⁼φ) → (◇⁼φ)⌋" using tBR by blast
lemma Incl_2:  "⌊(◇⁼φ) → (Eφ)⌋"       by blast
(*Section 3.6 "A binary preference fragment"*)
(* ≼EE is the dual of ≺AA *)
lemma "⌊(φ ≼EE ψ) ↔ ¬(ψ ≺AA φ)⌋ ∧ ⌊(φ ≺AA ψ) ↔ ¬(ψ ≼EE φ)⌋" by simp
(* ≼EE is the dual of ≺AA only if totality is assumed*)
lemma "⌊(φ ≼EE ψ) ↔ ¬(ψ ≺AA φ)⌋" nitpick oops (*countermodel*)
lemma "⌊(φ ≼EE ψ) → ¬(ψ ≺AA φ)⌋" by blast (*this direction holds*)
lemma "is_total SBR ⟶ ⌊(φ ≼EE ψ) ↔ ¬(ψ ≺AA φ)⌋"  by blast
lemma "⌊(φ ≺AA ψ) ↔ ¬(ψ ≼EE φ)⌋" nitpick oops (*countermodel*)
lemma "⌊(φ ≺AA ψ) → ¬(ψ ≼EE φ)⌋" by blast (*this direction holds*)
lemma "is_total SBR ⟶ ⌊(φ ≺AA ψ) ↔ ¬(ψ ≼EE φ)⌋"  by blast
(* verify p.97-98 *)
lemma monotonicity:  "⌊((φ ≼EE ψ) ∧ A(φ→ξ)) → (ξ ≼EE ψ)⌋" by blast
lemma reducibility:
      "⌊(((φ ≼EE ψ) ∧ α) ≼EE β) ↔ ((φ ≼EE ψ) ∧ (α ≼EE β))⌋" by blast
lemma reflexivity:   "⌊φ → (φ ≼EE φ)⌋" using rBR by blast
(*The condition below is supposed to enforce totality of the preference
  relation. However there are countermodels. See p.98?*)
lemma "is_total SBR ⟶
      ⌊((φ ≼EE φ)∧(ψ ≼EE ψ)) → ((φ ≼EE ψ)∨(ψ ≼EE φ))⌋"  by auto
lemma "⌊((φ ≼EE φ) ∧ (ψ ≼EE ψ)) → ((φ ≼EE ψ) ∨ (ψ ≼EE φ))⌋
       ⟶ is_total SBR" nitpick oops (*countermodel - error in paper?*)
lemma "is_total SBR ⟶
      ⌊((φ ≼EE φ) ∧ (ψ ≼EE ψ)) → ((φ ≼EE ψ) ∨ (ψ ≼EE φ))⌋" by auto
lemma "⌊((φ ≼EE φ) ∧ (ψ ≼EE ψ)) → ((φ ≼EE ψ) ∨ (ψ ≼EE φ))⌋
       ⟶ is_total SBR" nitpick oops (*countermodel - error in paper?*)
end
```

Figure 10: Experiments: Testing the meta-theory of $\mathcal{PL}$

We briefly explain the experiments shown in Fig. 10:

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 Benthem et al. (2009), except for the claims encoded in lines 41 and 42, where countermodels are reported by *Nitpick*.

Lines 25–47 Some application-specific tests in preparation for the modeling of the value ontology are conducted.

Further tests are shown in Fig. 11 for the *ceteris paribus* extension of $\mathcal{PL}$; we automatically prove all the relevant results from Benthem et al. (2009).

```
theory PreferenceLogicTests2      (*** Benzmüller & Fuenmayor, 2020 ***)
   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: "((E φ) w ⟷ (⟨∅⟩φ) w) ∧ ((A φ) w ⟷ ([∅]φ) w)" by auto
(**Axiomatization:**)
(*inclusion and interaction axioms *)
lemma Inc1: "⌊(⟨Γ⟩^≺φ) → (⟨Γ⟩^≼φ)⌋" 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 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" by auto
lemma "is_total SBR ⟶ (φ ≺AA^Γ ψ) u ⟷ (φ ≺AA^Γ ψ) u" by smt
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"  by auto
lemma "is_total SBR ⟶ (φ ≼AA^Γ ψ) u ⟷ (φ ≼AA^Γ ψ) u"  by smt
(*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 11: Experiments (cont'd): Testing the meta-theory of $\mathcal{PL}$

A.3 Testing the Value Ontology

Some further tests on the modeling and encoding of the value ontology are conducted; these tests are displayed in Fig. 12. Among others, we verify that the pair of operators for *extension* (↓) and *intension* (↑), cf. *Formal Concept Analysis* (Ganter and Wille 2012), constitute indeed a Galois connection (Lines 7–19), and we carry out some further tests on the value ontology (extending the ones presented in §5) concerning value aggregation and consistency (Lines 24ff.).

```
theory ValueOntologyTestLong (** Benzmüller, Fuenmayor & Lomfeld, 2020 **)
  imports ValueOntology
begin
lemma "True" nitpick[satisfy,show_all,card i=10] oops
lemma "⌊INCONSᴾ⌋" nitpick[satisfy,card i=4] nitpick oops (*contingent*)
(*ext/int 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 duala:  "(A₁ ⊔ A₂)↑ = (A₁↑ ⊓ A₂↑)" by blast
lemma dualb:  "(B₁ ⊔ B₂)↓ = (B₁↓ ⊓ B₂↓)" by blast
lemma         "(A₁ ⊓ A₂)↑ ⊑ (A₁↑ ⊔ A₂↑)" nitpick oops
lemma         "(B₁ ⊓ B₂)↓ ⊑ (B₁↓ ⊔ B₂↓)" nitpick oops
lemma dual2a: "(A₁↑ ⊔ A₂↑) ⊑  (A₁ ⊓ A₂)↑" by blast
lemma dual2b: "(B₁↓ ⊔ B₂↓) ⊑  (B₁ ⊓ B₂)↓" by blast
(*Note: two different but logically equivalent notations*)
lemma "[WILLˣ] ≡ WILLˣ↓" by simp
lemma "[WILLˣ⊕STABˣ] ≡ (WILLˣ⊕STABˣ)↓" by simp
(********* value ontology tests ****************)
lemma "⌊[RELIᴾ] ∧ [WILLᴾ] → INCONSᴾ⌋" by simp
lemma "⌊INCONSᴾ → [RELIᴾ] ∧ [WILLᴾ]⌋" by simp
lemma "⌊[RELIᴾ] ∧ [WILLᴾ]⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊[FAIRᵈ] ∧ [EFFIᵈ]⌋" nitpick[satisfy] nitpick oops (*contingent*)
lemma "⌊(¬INCONSᴾ) ∧ [FAIRᵈ] ∧ [EFFIᵈ]⌋"
 nitpick[satisfy,show_all] nitpick oops (*contingent: p & d independent*)
lemma "⌊(¬INCONSᵈ) ∧ (¬INCONSᴾ) ∧ [RELIᵈ] ∧ [WILLᴾ]⌋"
 nitpick[satisfy,show_all] nitpick oops (*contingent: p & d independent*)
(*** more tests ***)
(*values in two non-opposed quadrants (noq): consistent*)
lemma "⌊[WILLˣ] ∧ [STABˣ] → INCONSˣ⌋" nitpick oops (*countermodel found*)
lemma "⌊[WILLˣ] ∧ [GAINˣ] ∧ [EFFIˣ] ∧ [STABˣ] → INCONSˣ⌋" nitpick oops
(*values in two opposed quadrants: inconsistent*)
lemma "⌊[RESPˣ] ∧ [STABˣ] → INCONSˣ⌋" by simp
(*values in three quadrants: inconsistent*)
lemma "⌊[WILLˣ] ∧ [EFFIˣ] ∧ [RELIˣ] → INCONSˣ⌋" by simp
(*values in opposed quadrants for different parties: consistent*)
lemma "⌊[EQUIˣ] ∧ [GAINʸ] → (INCONSˣ ∨ INCONSʸ)⌋" nitpick oops (*cntmdl*)
lemma "⌊[RESPˣ] ∧ [STABʸ] → (INCONSˣ ∨ INCONSʸ)⌋" nitpick oops (*cntmdl*)
(*value preferences tests*)
lemma "⌊WILLˣ ≺ᵥ WILLˣ⊕STABˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊WILLˣ ≺ᵥ STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ WILLˣ⊕STABˣ⌋" by blast
lemma "⌊WILLˣ ≺ᵥ STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ RELIˣ⊕STABˣ⌋" by blast
lemma "⌊WILLˣ ≺ᵥ WILLˣ⊕STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ STABˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊WILLˣ ≺ᵥ RELIˣ⊕STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ STABˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
lemma "¬⌊WILLˣ⊕STABˣ ≺ᵥ WILLˣ⌋" using rBR by auto
lemma "⌊WILLˣ⊕STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊STABˣ ≺ᵥ WILLˣ⌋" by auto
lemma "⌊RELIˣ⊕STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊STABˣ ≺ᵥ WILLˣ⌋" by auto
lemma "⌊STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊WILLˣ⊕STABˣ ≺ᵥ WILLˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊RELIˣ⊕STABˣ ≺ᵥ WILLˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
(*basic properties*)
lemma "⌊¬(X ≺ᵥ X)⌋" using rBR by auto (*irreflexive*)
lemma "⌊X ≺ᵥ Y ⟶ ¬(Y ≺ᵥ X)⌋" nitpick oops (*not asymmetric*)
lemma "⌊(X ≺ᵥ Y ∧ Y ≺ᵥ Z) → X ≺ᵥ Z⌋" nitpick oops (*not transitive*)
end
```

Figure 12: Further testing of the value ontology

A.4 Modeling Conti v. ASPCA

Another illustrative case study we can model in our framework is: Conti v. ASPCA (cf. 353 NYS 2d 288).

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 (cf. §7) applies (with a preference in this case of FREEDOM over SECURITY, cf. §1). More specifically, it was ruled that for a domestic animal it is sufficient that the owner did not neglect or stopped caring for the animal, i.e. give up the responsability for its maintenance (RESP). This, together with ASPCA's reliance (RELI) in the parrot's property, outweighs Conti's corporal possession (STAB) of

the animal. A (simplified) argument is reconstructed as follows:

Let α stand for the animal. p and d stand for ASPCA and Conti respectively. ASPCA (p) argument goes like this:

1 ASPCA owned the parrot – (Prop p α).

2 The parrot escaped and was recaptured by Conti – (Capture d α).

3 However, ASPCA still cares for (recovering and maintaining) the animal – (Care p α).

4 Therefore, ruling for ASPCA – (For p) is preferred (since the combination of RELI & RESP is preferred to STAB by legal rule R3, cf. §7).

The reconstructed theory for the Conti vs. ASPCA case is displayed in Fig. 13.

```
theory Conti     (*** Benzmüller, Fuenmayor & Lomfeld, 2020 ***)
  imports GeneralKnowledge
begin (*** ASPCA v. Conti "wild animal" case **)

(*case-specific 'world-vocabulary'*)
consts α::"e" (*appropriated animal (parrot in this case) *)
consts Care::"c⇒e⇒σ"
consts Prop::"c⇒e⇒σ"
consts Capture::"c⇒e⇒σ"

(*case-specific taxonomic (legal domain) knowledge*)
axiomatization where
 CW1: "⌊Animal α ∧ Pet α → Domestic α⌋" and
 CW2: "⌊(∃c. Capture c α ∧ Domestic α) → appDomAnimal⌋" and
 CW3: "⌊∀c. Care c α → Mtn c⌋" and
 CW4: "⌊∀c. Prop c α → Own c⌋" and
 CW5: "⌊∀c. Capture c α → Poss c⌋"

lemma True nitpick[satisfy,card i=4] oops (*satisfiable*)

(***************** pro-ASPCA's argument ***************)
abbreviation "ASPCA_facts ≡ ⌊Parrot α ∧ Pet α ∧ Care p α ∧
                    Prop p α ∧ (¬Prop d α) ∧ Capture d α⌋"

(* decision for defendant (Conti) is compatible with premises*)
lemma "ASPCA_facts ∧ ⌊¬INCONSᵖ⌋ ∧ ⌊¬INCONSᵈ⌋ ∧ ⌊For p ≺ For d⌋"
  nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)

(* decision for plaintiff (ASPCA) is compatible with premises*)
lemma "ASPCA_facts ∧ ⌊¬INCONSᵖ⌋ ∧ ⌊¬INCONSᵈ⌋ ∧ ⌊For d ≺ For p⌋"
  nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)

(* decision for plaintiff (ASPCA) is provable*)
lemma aux: assumes ASPCA_facts shows "⌊(STABᵈ ≺ᵥ RELIᵖ⊕RESPᵖ)⌋"
  using CW1 CW2 W7 assms R3 by fastforce
theorem assumes ASPCA_facts shows "⌊For d ≺ For p⌋"
  using assms aux CW5 ForAx F3 other.simps(1) rBR by metis

(* while a decision for the defendant is refutable*)
lemma assumes ASPCA_facts shows "⌊For p ≺ For d⌋"
  nitpick[card i=4] oops (*(non-trivial) counterexample found*)
end
```

Figure 13: Modeling of the Conti vs. ASPCA case

A.5 Complex (Counter-)Models

An example of a complex countermodel with four possible worlds/states (type i) that has been computed by *Nitpick* for the statement in Line 41 in Fig. 13 is presented in Fig. 14.

This countermodel is included here to illustrate (to the interested reader) the richness of the information and the level of detail that is supported in our framework. It is in particular the combination of automated theorem proving, model finding and countermodel finding that is supporting the knowledge engineer and user of the LogiKEy framework in a unique manner to gain intuition about the modeled structures. And, in fact, these three analysis modes can be worked with in parallel in *Isabelle/HOL*. This is one reason for the good response rates to user requests that we often experience – despite the general undecidability of HOL.

```
Nitpick found a counterexample for card e = 1 and card i = 4:

  Skolem constant:
    λv. v = (λx. _)(i₁ := i₄, i₂ := i₁, i₃ := i₁, i₄ := i₄)
  Types:
    c = {d, p}
    e × i [boxed] = {(e₁, i₁), (e₁, i₂), (e₁, i₃), (e₁, i₄)}
    c VAL = {FREEDOM d, FREEDOM p, UTILITY d, UTILITY p, EQUALITY d, EQUALITY p, SECURITY d, SECURITY p}
  Constants:
    Capture =
      (λx. _)
      ((d, e₁, i₁) := True, (d, e₁, i₂) := True, (d, e₁, i₃) := True, (d, e₁, i₄) := True, (p, e₁, i₁) := False, (p, e₁, i₂) := False,
       (p, e₁, i₃) := False, (p, e₁, i₄) := False)
    Care =
      (λx. _)
      ((d, e₁, i₁) := False, (d, e₁, i₂) := False, (d, e₁, i₃) := False, (d, e₁, i₄) := True, (p, e₁, i₁) := True, (p, e₁, i₂) := True,
       (p, e₁, i₃) := True, (p, e₁, i₄) := True)
    Prop =
      (λx. _)
      ((d, e₁, i₁) := False, (d, e₁, i₂) := False, (d, e₁, i₃) := False, (d, e₁, i₄) := False, (p, e₁, i₁) := True, (p, e₁, i₂) := True,
       (p, e₁, i₃) := True, (p, e₁, i₄) := True)
    α = e₁
    Animal = (λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₁, i₃) := True, (e₁, i₄) := True)
    Domestic = (λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₁, i₃) := True, (e₁, i₄) := True)
    Fox = (λx. _)((e₁, i₁) := False, (e₁, i₂) := False, (e₁, i₃) := False, (e₁, i₄) := False)
    FreeRoaming = (λx. _)((e₁, i₁) := False, (e₁, i₂) := False, (e₁, i₃) := False, (e₁, i₄) := False)
    Intent =
      (λx. _)
      ((d, i₁) := False, (d, i₂) := True, (d, i₃) := False, (d, i₄) := True, (p, i₁) := False, (p, i₂) := False, (p, i₃) := True,
       (p, i₄) := False)
    Liv =
      (λx. _)
      ((d, i₁) := False, (d, i₂) := True, (d, i₃) := False, (d, i₄) := False, (p, i₁) := False, (p, i₂) := False, (p, i₃) := False,
       (p, i₄) := False)
    Mtn =
      (λx. _)
      ((d, i₁) := True, (d, i₂) := False, (d, i₃) := False, (d, i₄) := True, (p, i₁) := True, (p, i₂) := True, (p, i₃) := True,
       (p, i₄) := True)
    Own =
      (λx. _)
      ((d, i₁) := False, (d, i₂) := False, (d, i₃) := False, (d, i₄) := False, (p, i₁) := True, (p, i₂) := True, (p, i₃) := True,
       (p, i₄) := True)
    Parrot = (λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₁, i₃) := True, (e₁, i₄) := True)
    Pet = (λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₁, i₃) := True, (e₁, i₄) := True)
    Poss =
      (λx. _)
      ((d, i₁) := True, (d, i₂) := True, (d, i₃) := True, (d, i₄) := True, (p, i₁) := False, (p, i₂) := False, (p, i₃) := False,
       (p, i₄) := False)
    appAnimal = (λx. _)(i₁ := True, i₂ := True, i₃ := True, i₄ := True)
    appDomAnimal = (λx. _)(i₁ := True, i₂ := True, i₃ := True, i₄ := True)
    appObject = (λx. _)(i₁ := True, i₂ := True, i₃ := True, i₄ := True)
    appWildAnimal = (λx. _)(i₁ := False, i₂ := False, i₃ := False, i₄ := False)
    BR = (λx. _)
         ((i₁, i₁) := True, (i₁, i₂) := False, (i₁, i₃) := False, (i₁, i₄) := False, (i₂, i₁) := True, (i₂, i₂) := True,
          (i₂, i₃) := False, (i₂, i₄) := False, (i₃, i₁) := True, (i₃, i₂) := True, (i₃, i₃) := True, (i₃, i₄) := False,
          (i₄, i₁) := False, (i₄, i₂) := False, (i₄, i₃) := False, (i₄, i₄) := True)
    For =
      (λx. _)
      ((d, i₁) := False, (d, i₂) := True, (d, i₃) := True, (d, i₄) := False, (p, i₁) := True, (p, i₂) := False, (p, i₃) := False,
       (p, i₄) := True)
    ℐ = (λx. _)
        ((i₁, FREEDOM d) := True, (i₁, FREEDOM p) := True, (i₁, UTILITY d) := True, (i₁, UTILITY p) := False, (i₁, EQUALITY d) := False,
         (i₁, EQUALITY p) := True, (i₁, SECURITY d) := True, (i₁, SECURITY p) := True, (i₂, FREEDOM d) := True, (i₂, FREEDOM p) := True,
         (i₂, UTILITY d) := True, (i₂, UTILITY p) := False, (i₂, EQUALITY d) := True, (i₂, EQUALITY p) := False, (i₂, SECURITY d) := True,
         (i₂, SECURITY p) := True, (i₃, FREEDOM d) := False, (i₃, FREEDOM p) := True, (i₃, UTILITY d) := True, (i₃, UTILITY p) := False,
         (i₃, EQUALITY d) := True, (i₃, EQUALITY p) := True, (i₃, SECURITY d) := True, (i₃, SECURITY p) := False, (i₄, FREEDOM d) := True,
         (i₄, FREEDOM p) := True, (i₄, UTILITY d) := False, (i₄, UTILITY p) := False, (i₄, EQUALITY d) := True, (i₄, EQUALITY p) := True,
         (i₄, SECURITY d) := True, (i₄, SECURITY p) := True)
    other = (λx. _)(d := p, p := d)
```

Figure 14: Example of a countermodel to the statement in Line 41 in Fig. 13