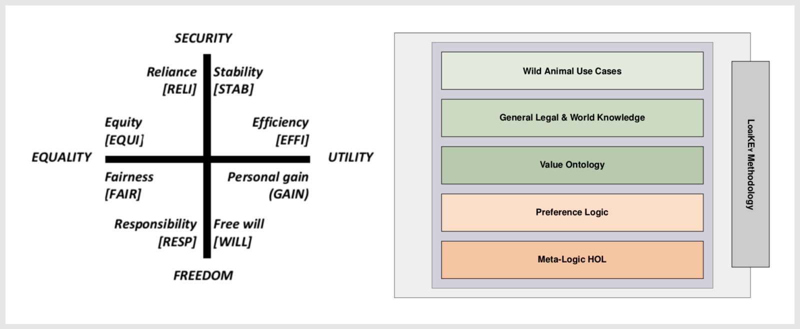
Encoding Legal Balancing: Automating an Abstract Ethico-Legal Value Ontology in Preference Logic

Christoph Benzmüller, David Fuenmayor, Bertram Lomfeld
Dep. of Mathematics and Computer Science & Dep. of Law
Freie Universität Berlin



Workshop "Deontic logic 2020-2030" (adapted from MLR@KR2020 talk)

Reasonable Machines: A Research Manifesto

Christoph Benzmüller & Bertram Lomfeld
Dep. of Mathematics and Computer Science & Dep. of Law, FU Berlin

Freie Universität Berlin

Abstract

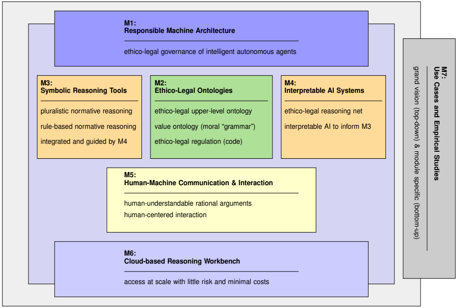
Future intelligent autonomous systems (IAS) are inevitably deciding on moral and legal questions, e.g. in self-driving cars, health care or human-machine collaboration. As decision processes in most modern sub-symbolic IAS are hidden, the simple political plea for transparency, accountability and governance falls short. A sound ecosystem of trust requires ways for IAS to autonomously justify their actions, that is, to learn giving and taking reasons for their decisions. Building on social reasoning models in moral psychology and legal philosophy such an idea of »Reasonable Machines« requires novel, hybrid reasoning tools, ethico-legal ontologies and associated argumentation technology. Enabling machines to normative communication creates trust and opens new dimensions of AI application and human-machine interaction.

Core Objectives

- enabling argument-based explanations & justifications of IAS decisions,
- enabling ethico-legal reasoning about, and public critique of, IAS decisions,
- facilitating political and legal governance of IAS decision making,
- evolving ethico-legal agency and communicative capacity of IASs,
- enabling trustworthy human-interaction by normative communication,
- fostering development of novel neuro-symbolic AI architectures.

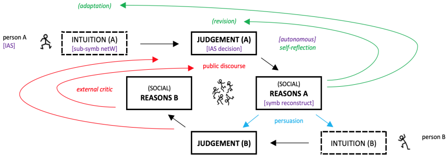
Long-term vision: To enable machines to give and take normative reasons for their decisions and actions capacitates them to engage in communicative action within social systems.

Modular structure of Reasonable Machines research



[Artificial] Social Reasoning Mechanism ([A]SRM)

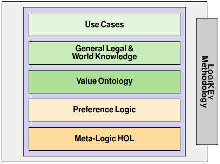
The parallel to human SRM guides the overall architectural design of Reasonable Machines.



For a legal SRM see e.g. [3]

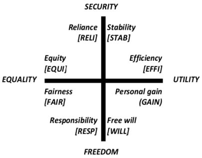
LogiKEy Methodology

LogiKEy [2] is a framework & methodology for the design and engineering of ethical reasoners, normative theories and deontic logics. For recent formalization work see [1] and logikey.org.



Value based Explanations

The reflexive symbolic/sub-symbolic feedback loop uses value categories based on an established discoursive moral grammar scheme [4], which we already encoded for legal balancing [1].



References

[1] Benzmüller, C., Fuenmayor, D., and Lomfeld, B. Encoding legal balancing: Automating an abstract ethico-legal value ontology in preference logic. In *WS on Models of Legal Reasoning, hosted by KR* (2020).

[2] Benzmüller, C., Parent, X., and van der Torre, L. Designing normative theories for ethical and legal reasoning: Logikey framework, methodology, and tool support. *Artificial Intelligence 287* (2020), 103348.

[3] Lomfeld, B. Emotio Iuris. Skizzen zu einer psychologisch aufgeklärten Methodenlehre des Rechts. In *Recht Fühlen* (2017), Köhler, Müller-Mall, Schmidt, and Schnädelbach, Eds., Fink, München, pp. 19–32.

[4] Lomfeld, B. Grammatik der Rechtfertigung: Eine kritische Rekonstruktion der Rechts(fort)bildung. *Kritische Justiz 52*, 4 (2019).

Motivation and Contribution

Bigger Vision:

- **Reasonable Machines: A Research Manifesto**
  (Benzmüller & Lomfeld, KI'2020, `http://dx.doi.org/10.1007/978-3-030-58285-2_20`)

Enabling machines to legal balancing

- Challenges: which logic? which value ontology? how to encode? interaction with other legal/world knowledge? which expressivity?
- LogiKEy-Solution: holistic, pluralistic framework; simultaneous modeling at different abstraction layers . . . until reflective equilibrium is reached

Main Contributions:

A: Universal (Meta-)Logical Reasoning and LogiKEy approach

- first-time application to support legal balancing
- first-time encoding of preference logic by vanBenthem et al.

B: Lomfeld's Value Ontology

- first-time operationalization on the computer
- in combination with preference logic by vanBenthem et al.

C: Combining A&B to model legal balancing in "Wild Animal Cases"

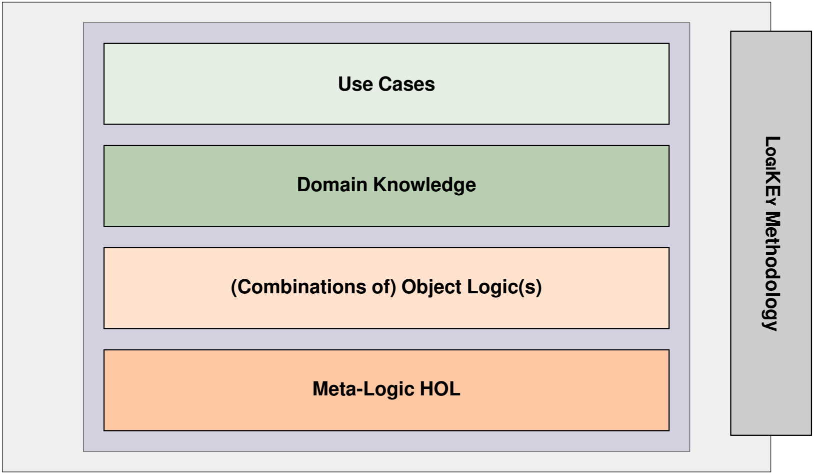
(A) Universal (Meta-)Logical Reasoning in HOL

[Science of Computer Programming (2019) vol. 172]

How to Tame the Logic Zoo?

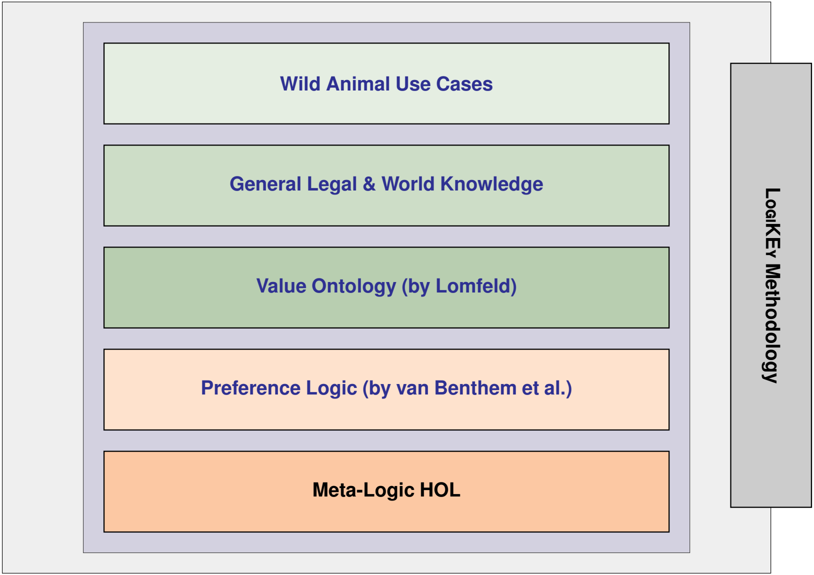
(A) LogiKEy Methodology

[Artificial Intelligence (2020) vol. 287]



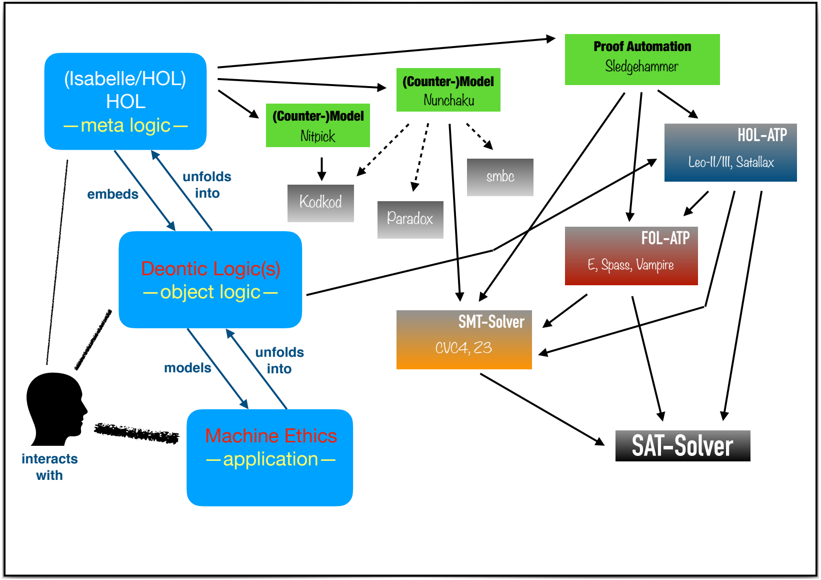
(A) LogiKEy Methodology

[Artificial Intelligence (2020) vol. 287]



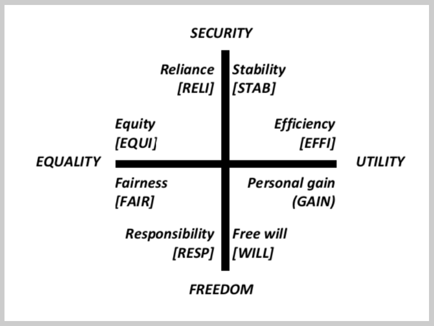
(A) Universal (Meta-)Logical Reasoning in Isabelle/HOL

[Science of Computer Programming (2019) vol. 172]



(B) Value Ontology and Preference Logic

Choice of Value Ontology:
Discoursive Grammar of Justification [Lomfeld (2015/2019)]



Choice of Formalization Logic:
(Modal) Logic for Preferences [vanBenthemGirardRoy(2009), JPL]

SpringerLink

Published: 13 August 2008

Everything Else Being Equal: A Modal Logic for *Ceteris Paribus* Preferences

Johan van Benthem, Patrick Girard & Olivier Roy

Journal of Philosophical Logic **38**, 83–125(2009) | Cite this article

393 Accesses | **75** Citations | **0** Altmetric | Metrics

(B) "Discourse Logic" of Legal Balancing

- Legal reasoning is seen as practical argumentation with a two-level model of (more abstract) values & principles and (more concrete) legal rules.
- Legal rules (or common-law precedents) can be reconstructed as *conditional* preference relations between conflicting underlying value principles (cf. Alexy 2000; Lomfeld 2015)

Example: *"In view of events E_1 (a virus pandemic occurs) and E_2 (voluntary shut-down fails) countrywide lock-down becomes sanctioned, since health security outweighs freedom to move."*

Application of a rule **R** involves balancing value principles **A** (SECURITY) and **B** (FREEDOM) *in context* (conditions E_1 and E_2):

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

Acts as justification for the rule's legal consequence (e.g. sanctioned lock-down).

(B) Encoding using a Logic of Preferences:

Choice of Formalization Logic:
[vanBenthemGirardRoy(2009)JPL]

Published: 13 August 2008

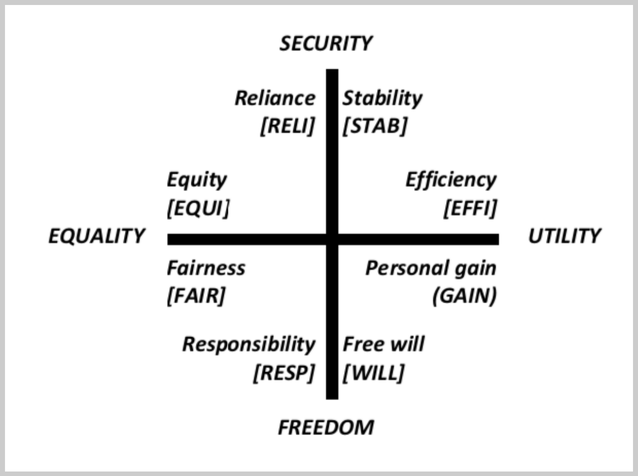
Everything Else Being Equal: A Modal Logic for *Ceteris Paribus* Preferences

Johan van Benthem, Patrick Girard & Olivier Roy

Journal of Philosophical Logic **38**, 83–125(2009) | Cite this article

(B) Value Ontology

But which value principles are to be balanced? [Lomfeld (2015), (2019)]



- In our case studies: a decision promoting a particular value (over others) corresponds to ruling for a certain party. (Values are indirectly 'assigned' to particular parties/actors using 'factors'.)

(B) Value Ontology

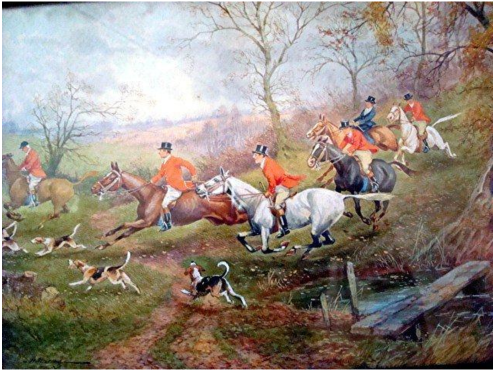
Comparison between some relevant value-based approaches in the literature [Lomfeld (2020)]

VALUES & legal principles	Berman & Hafner 1993	Bench-Capon et al 2005	Bench-Capon 2012	Gordon & Walton 2012	Prakken 2002	Sartor 2002 (Sartor 2010)
FREEDOM - Free choice (WILL) - Responsibility (RESP)	"Protect from interference"	"Court should not make law"	"Reward"			("Liberty")
SECURITY - Stability (STAB) - Reliance (RELI)	"Certainty"	"Clear law"	"Legal certainty" "Public order"	"Security"	"Legal certainty"	("Security") "Less litigation" "Sec. possession"
EQUALITY - Fairness (FAIR) - Equity (EQUI)	"Property rights" "Public land"	"Property"	"Fairness"	"Fairness" "Equity"	"Property rights"	
UTILITY - Efficiency (EFFI) - Personal gain (GAIN)	"Free enterprise and competition"	"Useful" & "Economic activity"	"Utility" - "Econ. valuable" - "Personal gain"		"Economic benefit for society"	"Productivity"

Ambition:

- To consistently cover existing value sets from formal argumentation and AI & Law accounts on value-based reasoning, e.g. (Berman and Hafner 1993; Bench-Capon 2012; Gordon and Walton 2012; Sartor 2010).

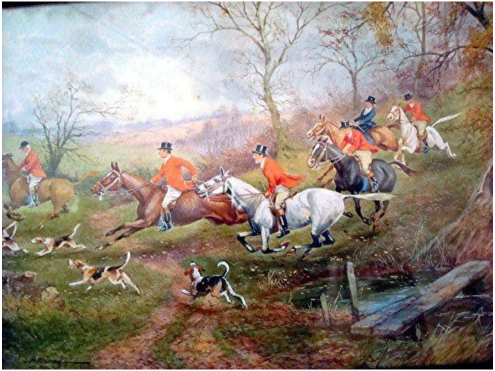
(C) Case Study: Pierson vs. Post



Maybe the most famous property law case in American legal history:

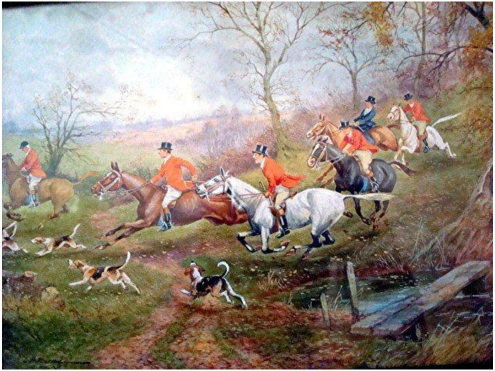
Post, a fox hunter, was chasing a fox through public land when Pierson came across the fox and, knowing it was being chased, killed the fox and took it away. Post sued Pierson for damages against his possession of the fox. Post argued that giving chase to the fox was sufficient to establish possession.

(C) Case Study: Pierson vs Post



- A local court first ruled in favour of Post.
- However, Pierson appealed the ruling to the New York Supreme Court of Judicature, who reversed the decision
- The court ruled in favor of Pierson; citing ancient and modern precedents: "pursuit alone vests no property" (Justinian); and "corporal possession creates legal certainty" (Pufendorf).

(C) Case Study: Pierson vs Post



In our framework:

- The decision in favour of Pierson implies: STAB(ility) $>$ WILL.
- **For "wild animal cases":** the legal certainty created by corporal possession (STAB) has preference over "pursuit alone" (WILL).
- Notice the context of validity for the value preference above. Alternatively, Post might argue against this being a "wild animal case".

(C) Case Study: Conti vs ASPCA

Another famous property law case concerning (wild?) animals:

Chester, a parrot owned by the ASPCA (animal shelter), escaped and was recaptured by Conti. The ASPCA found this out and reclaimed Chester from Conti.

(C) Case Study: Conti vs ASPCA

In this case, the court ruled in favour of the ASPCA:

- For **domestic animals** the value preference relation as in Pierson's case does not apply,
- 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/B/C: Demo

Isabelle/HOL Encodings&Tests

- ▶ Preference Logic
- ▶ Preference Logic Tests
- ▶ Value Ontology
- ▶ Value Ontology Tests
- ▶ General (World) Knowledge
- ▶ Pierson Case
- ▶ Conti Case

```
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)"
```

Following [vanBenthemGirardRoy(2009)JPL]

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- ► Preference Logic
- ► Preference Logic Tests
- ► Value Ontology
- ► Value Ontology Tests
- ► General (World) Knowledge
- ► Pierson Case
- ► Conti Case

```
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*)
(**** 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
```

Following [vanBenthemGirardRoy(2009)JPL]

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case

Following [vanBenthemGirardRoy(2009)JPL]

```
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
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case

Following [vanBenthemGirardRoy(2009)JPL]

```
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: "((Eφ) w ⟷ (⟨∅⟩φ) w) ∧ ((Aφ) 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
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- ▶ Preference Logic
- ▶ Preference Logic Tests
- ▶ Value Ontology
- ▶ Value Ontology Tests
- ▶ General (World) Knowledge
- ▶ Pierson Case
- ▶ Conti Case

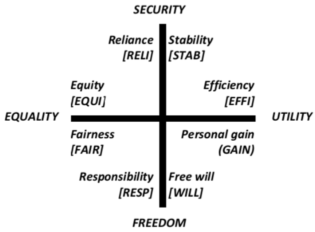
Following [vanBenthemGirardRoy(2009)JPL]

```
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
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



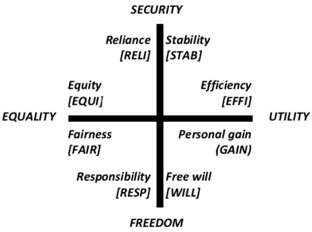
Following [Lomfeld(2019)KritischeJustiz]

```
theory ValueOntology      (*** Benzmüller, Fuenmayor & Lomfeld, 2020 ***)
  imports PreferenceLogicBasics
begin (*** Lomfeld's value ontology is encoded ***)

(*two legal parties (there can be more in principle)*)
datatype c = p | d (*parties/contenders: plaintiff, defendant*)
fun other::"c⇒c" ("_^-1") where "p^-1 = d" | "d^-1 = p"

consts For::"c⇒σ"    (*decision: find/rule for party*)
axiomatization where ForAx: "⌊For x ↔ (¬For x^-1)⌋"

datatype (*ethico-legal upper values (wrt. a given party)*)
 't VAL = FREEDOM 't | UTILITY 't | SECURITY 't | EQUALITY 't
type_synonym v = "(c)VAL⇒bool" (*principles: sets of upper values*)
type_synonym cv = "c⇒v" (*principles are specified wrt. a given party*)

abbreviation vset1 ("⦃_⦄") where "⦃φ⦄ ≡ λx::(c)VAL. x=φ"
abbreviation vset2 ("⦃_,_⦄")   where "⦃α,β⦄ ≡ λx::(c)VAL. x=α ∨ x=β"

abbreviation utility::cv  ("UTILITY_")  where "UTILITY^x  ≡ ⦃UTILITY  x⦄"
abbreviation security::cv ("SECURITY_") where "SECURITY^x ≡ ⦃SECURITY x⦄"
abbreviation equality::cv ("EQUALITY_") where "EQUALITY^x ≡ ⦃EQUALITY x⦄"
abbreviation freedom::cv  ("FREEDOM_")  where "FREEDOM^x  ≡ ⦃FREEDOM  x⦄"
abbreviation stab::cv ("STAB_") where "STAB^x ≡ ⦃SECURITY x, UTILITY  x⦄"
abbreviation effi::cv ("EFFI_") where "EFFI^x ≡ ⦃UTILITY  x, SECURITY x⦄"
abbreviation gain::cv ("GAIN_") where "GAIN^x ≡ ⦃UTILITY  x, FREEDOM  x⦄"
abbreviation will::cv ("WILL_") where "WILL^x ≡ ⦃FREEDOM  x, UTILITY  x⦄"
abbreviation resp::cv ("RESP_") where "RESP^x ≡ ⦃FREEDOM  x, EQUALITY x⦄"
abbreviation fair::cv ("FAIR_") where "FAIR^x ≡ ⦃EQUALITY x, FREEDOM  x⦄"
abbreviation equi::cv ("EQUI_") where "EQUI^x ≡ ⦃EQUALITY x, SECURITY x⦄"
abbreviation reli::cv ("RELI_") where "RELI^x ≡ ⦃SECURITY x, EQUALITY x⦄"

(*derivation operators (cf. theory of "formal concept analysis") *)
consts Vrel::"i⇒(c)VAL⇒bool" ("ℐ") (*incidence relation worlds-values*)
abbreviation intension::"σ⇒v" ("_↑") where "W↑ ≡ λv. ∀x. W x ⟶ ℐ x v"
abbreviation extension::"v⇒σ" ("_↓") where "V↓ ≡ λw. ∀x. V x ⟶ ℐ w x"

(*shorthand notation for aggregating values*)
abbreviation agg (infixr "⊕"80) where "v1⊕v2 ≡ v1 ⊓ v2"
abbreviation agg1 ("[_]") where "[v] ≡ v↓"
abbreviation agg2 ("[_⊕_]") where "[v1⊕v2] ≡ (v1⊕v2)↓"
abbreviation agg3 ("[_⊕_⊕_]") where "[v1⊕v2⊕v3] ≡ (v1⊕v2⊕v3)↓"
abbreviation agg4 ("[_⊕_⊕_⊕_]") where "[v1⊕v2⊕v3⊕v4] ≡ (v1⊕v2⊕v3⊕v4)↓"

(*chosen variant for preference relation (cf. van Benthem et al. 2009*)
abbreviation relPref::"σ⇒σ⇒σ"  ("_≺_") where "φ ≺ ψ ≡ ψ ≻EA φ"
abbreviation relPrefval::"v⇒v⇒σ" ("_≺v_") where "φ ≺v ψ ≡ ψ↓ ≻EA φ↓"

abbreviation incnst ("INCONS_") where (*inconsistency for value support*)
 "INCONS^x ≡ [SECURITY^x] ∧ [EQUALITY^x] ∧ [FREEDOM^x] ∧ [UTILITY^x]"
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



Following [Lomfeld(2019)KritischeJustiz]

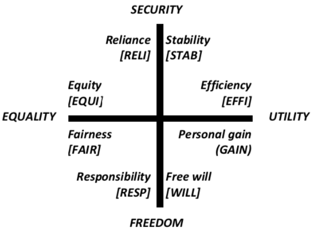
```
theory ValueOntologyTest (*** Benzmüller, Fuenmayor & Lomfeld, 2020 ***)
  imports ValueOntology
begin (* value ontology tests *)
(*values in two opposed quadrants: inconsistent*)
lemma "⌊[RESPˣ] ∧ [STABˣ] → INCONSˣ⌋" by simp
lemma "⌊[RELIˣ] ∧ [WILLˣ] → INCONSˣ⌋" by simp
(*all values in two non-opposed quadrants: consistent*)
lemma "⌊[WILLˣ] ∧ [STABˣ] → INCONSˣ⌋" nitpick oops  (*countermodel*)
(*values in opposed quadrants for different parties: consistent*)
lemma "⌊[EQUIˣ] ∧ [GAINʸ] → (INCONSˣ ∨ INCONSʸ)⌋" nitpick oops (*ctm*)
lemma "⌊[RESPˣ] ∧ [STABʸ] → (INCONSˣ ∨ INCONSʸ)⌋" nitpick oops (*ctm*)
lemma "⌊[RELIᵖ] ∧ [WILLᵖ]⌋" nitpick[satisfy] nitpick oops (*contingent*)
(*value preferences tests*)
lemma "⌊WILLˣ ≺ᵥ STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ RELIˣ⊕STABˣ⌋" by blast
lemma "⌊RELIˣ⊕STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊STABˣ ≺ᵥ WILLˣ⌋" by auto
lemma "⌊WILLˣ ≺ᵥ RELIˣ⊕STABˣ⌋ ⟶ ⌊WILLˣ ≺ᵥ STABˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
lemma "⌊STABˣ ≺ᵥ WILLˣ⌋ ⟶ ⌊RELIˣ⊕STABˣ ≺ᵥ WILLˣ⌋"
  nitpick nitpick[satisfy] oops (*contingent*)
end
```

```
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)
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



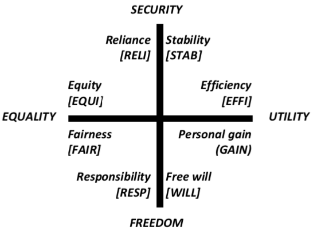
Following [Lomfeld(2019)KritischeJustiz]

```
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 dual1a: "(A₁ ⊔ A₂)↑ = (A₁↑ ⊓ A₂↑)" by blast
lemma dual1b: "(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
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



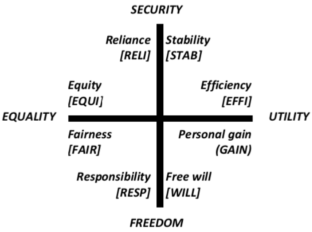
Following [Lomfeld(2019)KritischeJustiz]

```
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*)
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



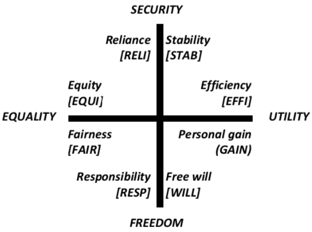
Following [Lomfeld(2019)KritischeJustiz]

```
(*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)  → (STABˣ⁻¹ ≺ᵥ RESPˣ⊕RELIˣ)⌋"
 (*...others...*)

(*LWK: relate values, outcomes and situational 'factors'*)
axiomatization where
 F1: "⌊For x → (Intent x ↔ □≼[WILLˣ])⌋" and
 F2: "⌊For x → (Mal x⁻¹ ↔ □≼[RESPˣ])⌋" and
 F3: "⌊For x → (Poss x ↔ □≼[STABˣ])⌋" and
 F4: "⌊For x → (Mtn x ↔ □≼[RESPˣ])⌋" and
 F5: "⌊For x → (Own x ↔ □≼[RELIˣ])⌋"

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

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



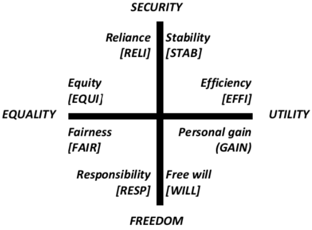
Following [Lomfeld(2019)KritischeJustiz]

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

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

(*case-specific taxonomic (legal domain) knowledge*)
axiomatization where
 CW1: "⌊(∃c. Capture c α ∧ ¬Domestic α)  → appWildAnimal⌋" and
 CW2: "⌊∀c. Pursue c α → Intent c⌋" and
 CW3: "⌊∀c. Capture c α → Poss c⌋"

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

(***************** pro-Pierson's argument ***************)
abbreviation "Pierson_facts ≡ ⌊Fox α ∧ (FreeRoaming α) ∧
  (¬Pet α) ∧ Pursue p α ∧ (¬Pursue d α) ∧ Capture d α⌋"

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

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

(*decision for defendant (Pierson) is provable*)
theorem assumes Pierson_facts shows "⌊For p ≺ For d⌋"
  by (metis assms CW1 CW2 W6 W8 ForAx R2 F1 other.simps(2) rBR)

(*while a decision for the plaintiff is not*)
lemma assumes Pierson_facts shows "⌊For d ≺ For p⌋"
  nitpick[card i=4] oops (*counterexample found*)

(***************** pro-Post's argument ***************)
(* Theory amendment: the animal is not free-roaming since it
   is being chased by a professional hunter (Post) *)
consts Hunter::"c⇒σ"
axiomatization where (*case-specific legal rule for hunters*)
 R5: "⌊(Hunter x ∧ Pursue x α)  → (STABˣ⁻¹ ≺ᵥ EFFIˣ)⌋"

abbreviation "Post_facts ≡ ⌊Fox α ∧ (¬FreeRoaming α) ∧
   Hunter p ∧ Pursue p α ∧ (¬Pursue d α) ∧ Capture d α⌋"
```

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



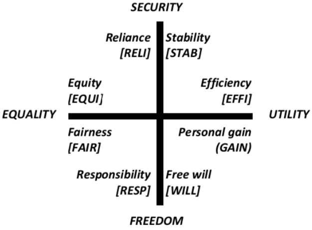
Following [Lomfeld(2019)KritischeJustiz]

```
abbreviation "Pierson_facts ≡ ⌊Fox α ∧ (FreeRoaming α) ∧
  (¬Pet α) ∧ Pursue p α ∧ (¬Pursue d α) ∧ Capture d α⌋"

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

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

(*decision for defendant (Pierson) is provable*)
theorem assumes Pierson_facts shows "⌊For p ≺ For d⌋"
  by (metis assms CW1 CW2 W6 W8 ForAx R2 F1 other.simps(2) rBR)

(*while a decision for the plaintiff is not*)
lemma assumes Pierson_facts shows "⌊For d ≺ For p⌋"
  nitpick[card i=4] oops (*counterexample found*)

(***************** pro-Post's argument ***************)
(* Theory amendment: the animal is not free-roaming since it
   is being chased by a professional hunter (Post) *)
consts Hunter::"c⇒σ"
axiomatization where (*case-specific legal rule for hunters*)
 R5: "⌊(Hunter x ∧ Pursue x α)  → (STAB^x⁻¹ ≺_v EFFI^x)⌋"

abbreviation "Post_facts ≡ ⌊Fox α ∧ (¬FreeRoaming α) ∧
   Hunter p ∧ Pursue p α ∧ (¬Pursue d α) ∧ Capture d α⌋"

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

(*decision for plaintiff (Post) is compatible with premises too*)
lemma "Post_facts ∧ ⌊¬INCONS^p⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊For d ≺ For p⌋"
  nitpick[satisfy,card i=4] oops (* (non-trivial) model found*)

(*indeed, a decision for plaintiff (Post) now becomes provable*)
theorem assumes Post_facts shows "⌊For d ≺ For p⌋"
  using assms by (metis CW3 ForAx R5 F3 other.simps rBR)

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

A/B/C: Demo

Isabelle/HOL Encodings&Tests

- Preference Logic
- Preference Logic Tests
- Value Ontology
- Value Ontology Tests
- General (World) Knowledge
- Pierson Case
- Conti Case



Following [Lomfeld(2019)KritischeJustiz]

```
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 ∧ ⌊¬INCONS^p⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊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 ∧ ⌊¬INCONS^p⌋ ∧ ⌊¬INCONS^d⌋ ∧ ⌊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 "⌊(STAB^d ≺_v RELI^p⊕RESP^p)⌋"
  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
```

Models and Countermodels are particularly helpful!

```
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)
```

Conclusion and Related Work

Contributions:

- Feasibility study for **legal balancing on the computer**
- Embedding of Preference Logic in HOL
- Demonstrated formalization&use of **Lomfeld's value ontology**
- Successful application of
  – LogiKEy methodology and
  – Universal (Meta-)Logical Reasoning in HOL
- **Flexibility**, **Expressiveness** and ready to use **ATP Support**!

Related work:

- Constructive interpretation in law, including model of value balancing: [Maranhão&Sartor(2019)ICAIL]
- Models to quantify legal balancing: [Alexy(2003), Sartor(2010)]

Bigger Vision:

Reasonable Machines: A Research Manifesto
(Benzmüller & Lomfeld, KI'2020, `http://dx.doi.org/10.1007/978-3-030-58285-2_20`)