Legal Reasoning With Argumentation Schemes

Thomas F. Gordon
Fraunhofer FOKUS
Berlin, Germany

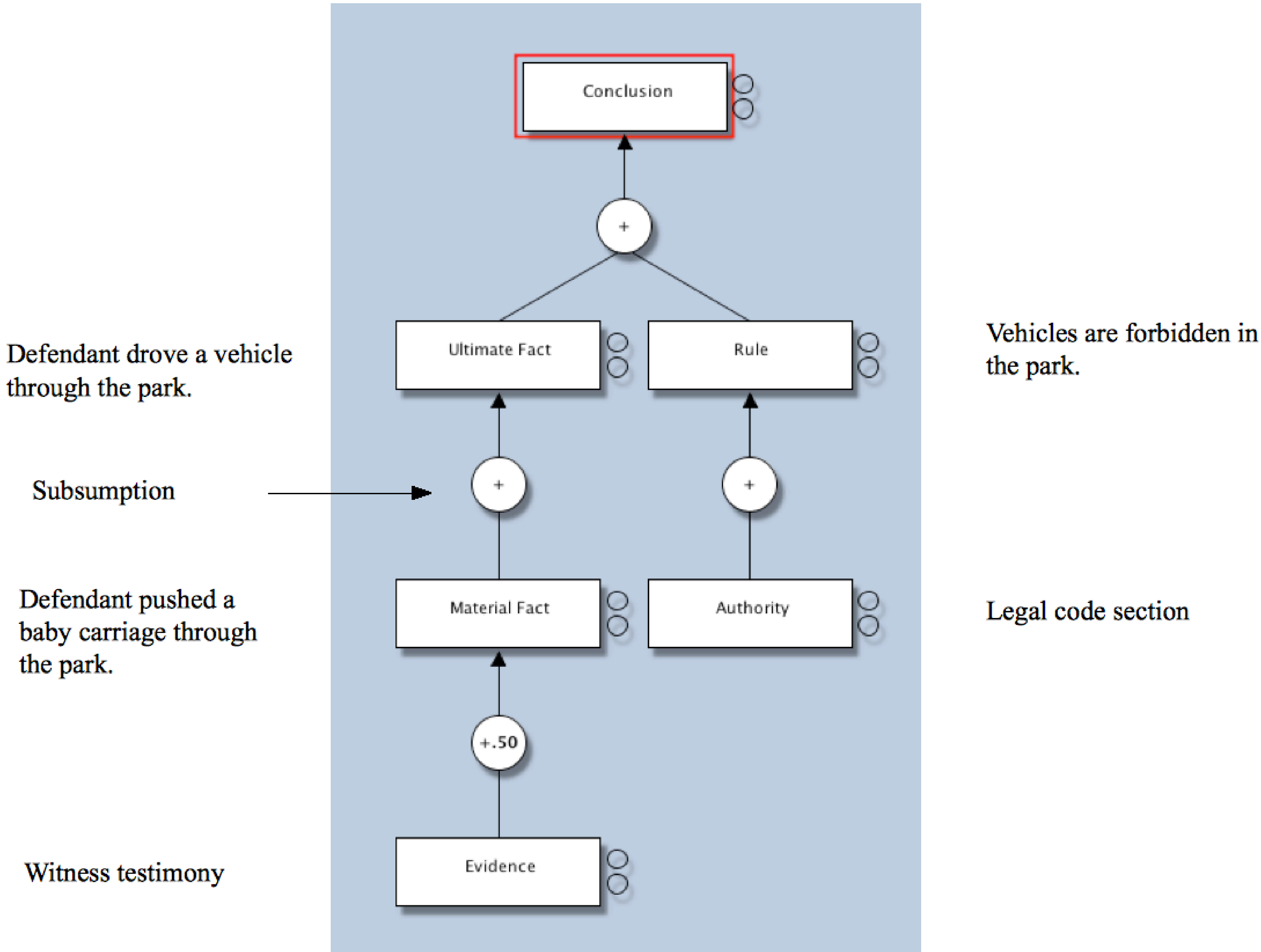
thomas.gordon@fokus.fraunhofer.de

Douglas Walton
University of Windsor
Centre for Research on Reasoning,
Argumentation and Rhetoric
Windsor, Canada

dwalton@uwindsor.ca

Legal Reasoning

- Is not only deduction from rules and facts
- But rather also a **modeling** process [Fiedler, 1985] in which
  - **Theories** of the law and fact are **constructed**, and
  - **Arguments are constructed** from these theories, for both sides of the issue.
- That is, theories of the law and facts are not given, a priori, but one result of the process.
- Theories of the law and facts of a case are interdependent and need to be constructed together [Engisch, 1960]



Conclusion
+
Defendant drove a vehicle through the park.
Ultimate Fact
Rule
Vehicles are forbidden in the park.
Subsumption
+
+
Defendant pushed a baby carriage through the park.
Material Fact
Authority
Legal code section
+.50
Witness testimony
Evidence

Argumentation Schemes

Two complementary views:

- **Patterns of argument**, used e.g. to classify arguments and reveal missing or faulty premises
- **Methods for constructing or "inventing" arguments** which instantiate these patterns

Some Kinds of Argumentation Schemes Useful for Legal Reasoning

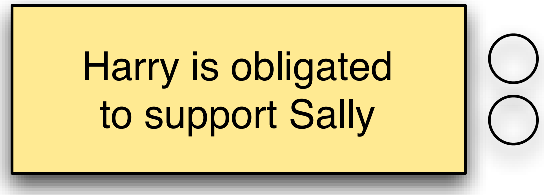
- Argument from legal rules
- Argument from concepts (ontology)
- Argument from cases or theories of a series of cases
- Argument from evidence, especially testimony
- Argument from legal principles
- Argument from ethics
- Argument from policy (teleology)

A Protocol for Hybrid Reasoning Using Argumentation Schemes

- **Argument generators** are argumentation schemes in their role as methods for constructing arguments
- An **argument generator** is a function which given:
  - A set of arguments
  - A set of assumptions, and
  - A statement at issue

  produces a **stream of arguments** pro or con the statement at issue.
- **Key Idea**: Various hybrid reasoning methods can be integrated and used together by wrapping them with a layer which implements this common argument generator protocol.

An Example to Guide Us



Argument from Legal Rules

- Rules are “reified” objects with properties, e.g. date of enactment.
- Rules are subject to exceptions.
- Rules can conflict.
- Some conflicts can be resolved using rules about rule priorities, e.g. lex superior.
- Rules can be excluded from being applicable by other rules
- Rules can be invalid. Deleting invalid rules from the KB is not an option.
- There is much consensus in AI and Law about these features [Gordon 1993; Hage 1993; Prakken & Sartor, 1996]

Scheme for Arguments from Rules

Premises

- R is a legal rule with antecedents A1, …, An and conclusion C.
- Each Ai in A1, ..., An is presumably true.

Conclusion

- C is presumably true.

Critical Questions

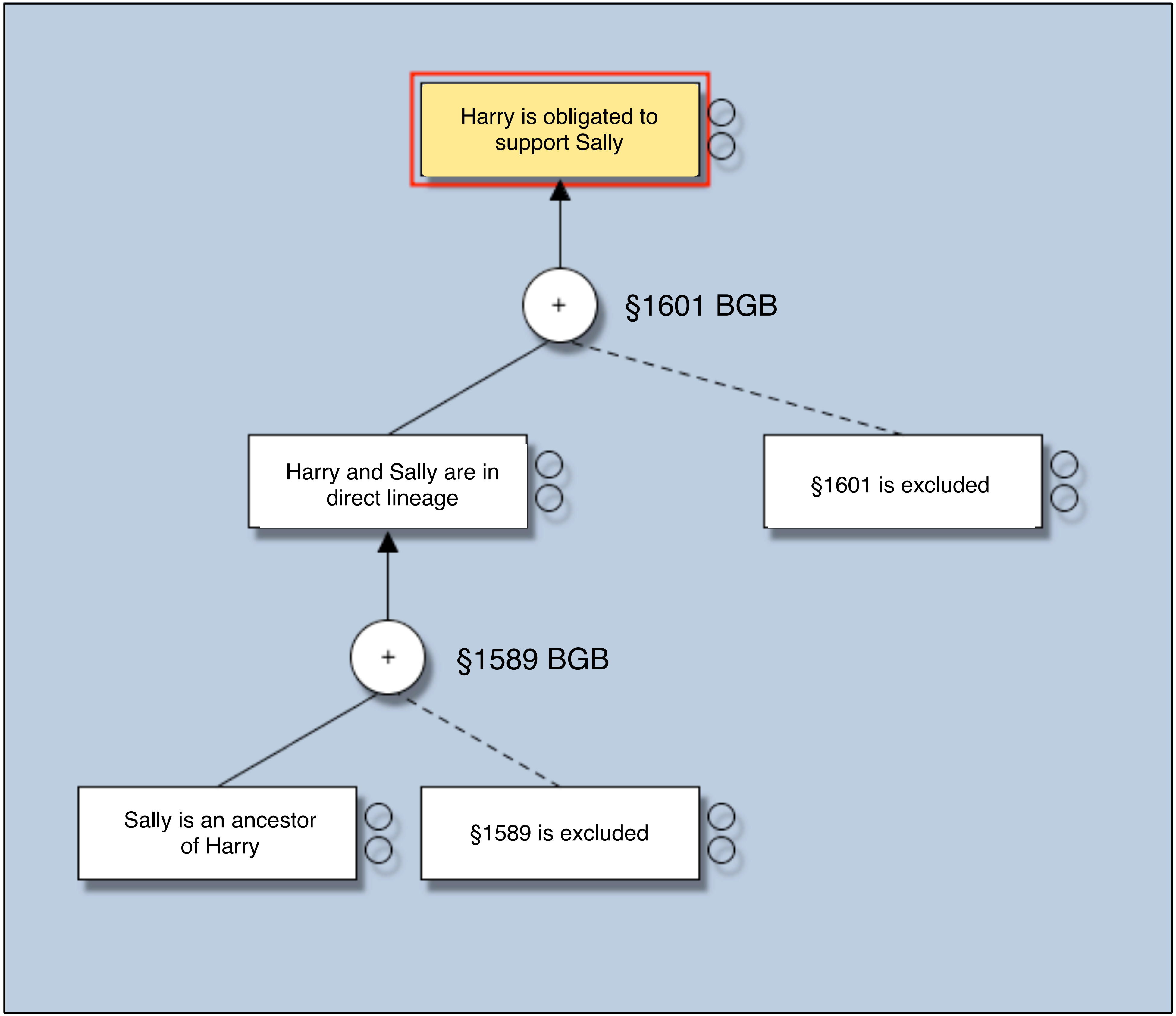
- Does some **exception** to R apply?
- Is some **assumption** of R not met?
- Is R a **valid** legal rule?
- Does some rule **excluding** R apply in this case?
- Does some conflicting rule of **higher priority** than R apply in this case?

Example Rules, Modeled in Legal Knowledge Interchange Format (LKIF)

rule s1601-BGB.
Person1 is obligated to support Person2
given
Person1 is in direct lineage to Person2

rule s1589a-BGB.
Person1 is in direct lineage to Person2
given
Person1 is an ancestor of Person2

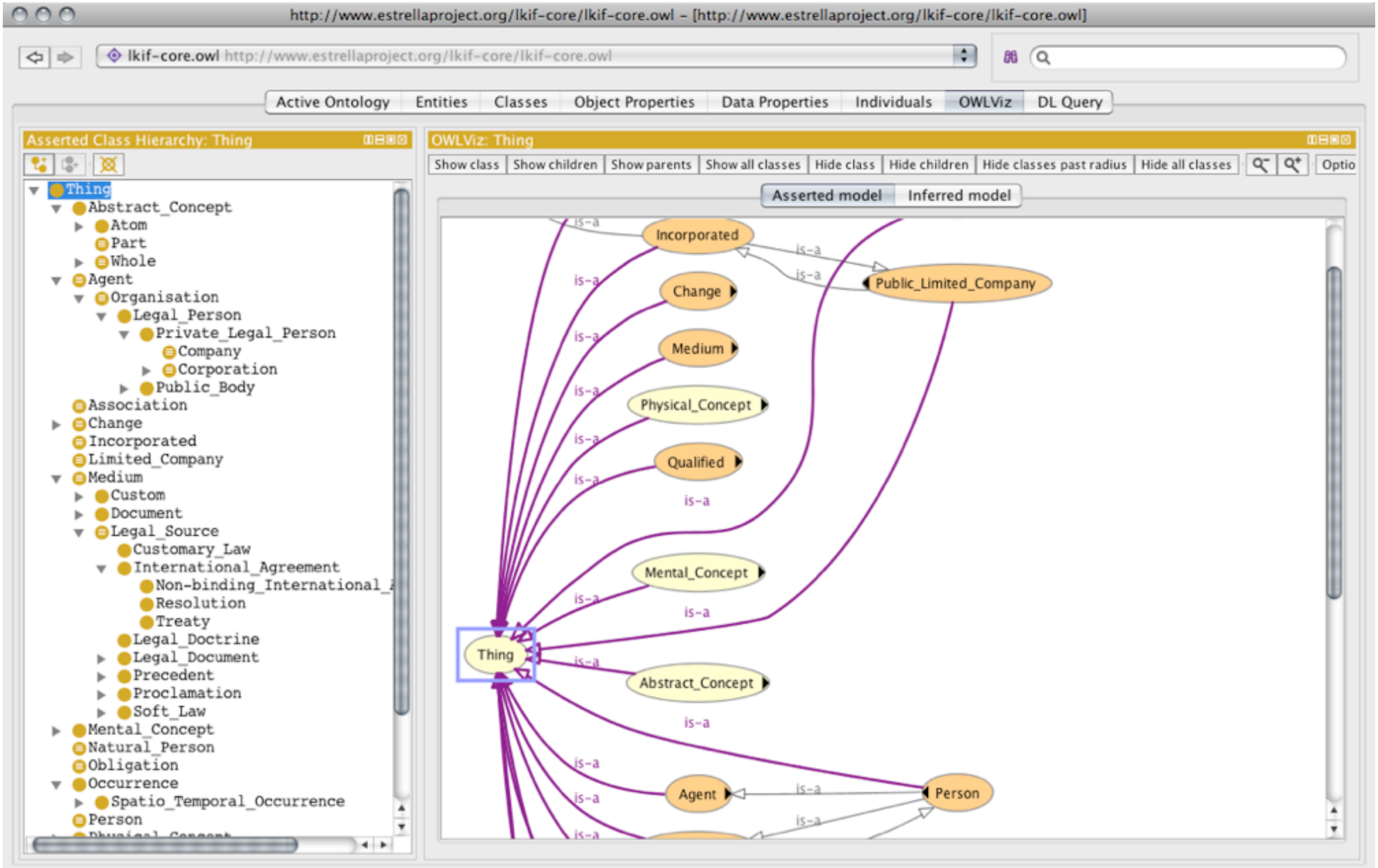
rule s91-BSHG.
s1601-BGB excludes "Person1 is obligated to support Person2"
given
"Person1 is obligated to support Person2" would cause Person1 undue hardship


Harry is obligated to support Sally
§1601 BGB
Harry and Sally are in direct lineage
§1601 is excluded
§1589 BGB
Sally is an ancestor of Harry
§1589 is excluded

Argument from Ontology

- In computer science, an “ontology” is an advanced kind of entity-relationship data model.
- Ontologies are used to standardize the semantics of data models, to facilitate the interchange of data among programs, abstracting away syntactic and other details.
- Important role of ontologies in our architecture for hybrid reasoning: They enable a common terminology across different models and reasoners.

Example: ESTRELLA's Ontology of Basic Legal Concepts

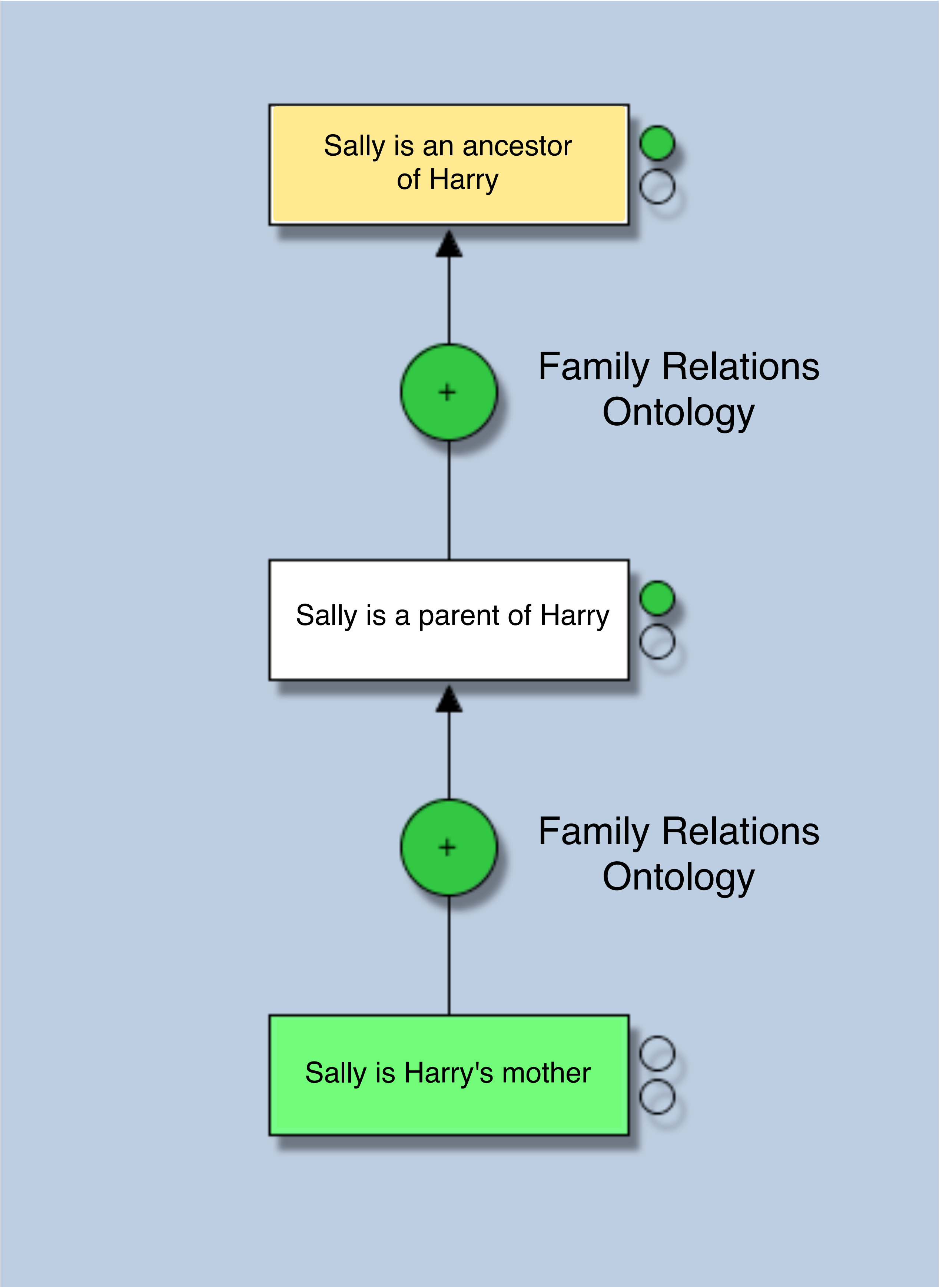


Description Logic Programming (DLP)

Description Logic	Predicate Logic	Meaning	Example
$C \subseteq D$ where C and D are classes (concepts)	$D(x) \leftarrow C(x)$	Cs are Ds. C is a subclass of D.	Peguins are birds.
$Q \subseteq P$ where Q and P are properties (roles)	$P(x,y) \leftarrow Q(x,y)$	Qs of x are Ps of x. Q is a subproperty of P.	The mother of a person is a parent of the person.
$\forall R . C$	$C(y) \leftarrow R(x,y)$	Every R of x is a C. The range of R is C.	The mother of a person is a woman.
$C \subseteq \exists R . D$	$C(x) \leftarrow R(x,y) \wedge D(y)$	Objects which have an R which is a D is a C.	Persons who own a home in Bel Air are wealthy.
$C \cap D \subseteq E$	$E(x) \leftarrow C(x) \wedge D(x)$	Instances of both C and D are also instances of E.	Anything which is male and human is a man.
$C \subseteq D \cap E$	$D(x) \leftarrow C(x)$ $E(x) \leftarrow C(x)$	Instances of C are also instances of both D and E.	Every woman is human and female.

Argumentation Scheme for DLP Ontologies

- Our approach, using the correspondence between DLP and Predicate Logic, is to:
    1. Translate the ontology into rules
    2. Use the scheme for arguments from rules
- Disadvantage: Limited to the DLP dialect of Description Logic. (More expressive Description Logics exist.)


Sally is an ancestor of Harry
Family Relations Ontology
Sally is a parent of Harry
Family Relations Ontology
Sally is Harry's mother

Are Arguments from Ontology Defeasible?

- Argument from ontology is a special kind of argument from theory, using only the terminological axioms of the theory

- Argument from theory
  - derivability premise: $T \vdash P$
  - theory premise: T is a coherent theory of the intended domain.
  - conclusion: P

- Critical Questions
  - Even though P is necessarily true if T is true, the argument can be challenged by questioning the theory premise. Is the theory T really coherent?

- Thus, in our view, the conclusion of an argument from ontology is, like all arguments, only conditionally and presumptively true.

Arguments from Cases

- TAXMAN II [McCarty & Sridharan 1981] – First to model argument from theories, using prototypes and deformations of concepts in cases.
- HYPO [Ashley & Rissland, 1990] – Modeled arguments from analogy with factor comparison
- CABARET [Skalak & Rissland, 1991] – Used cases to broaden and narrow the interpretation of rules
- GREBE [Branting 1991] - Used rules to match cases and cases to satisfy open-textured concepts in rules.
- CATO [Aleven & Ashley 1997] - Introduced factor hierarchies to support arguments from downplaying and emphasizing case distinctions.
- Bench-Capon & Sartor [2003] used social values to construct theories of cases.

Example HYPO Argumentation Schemes

- Cite Analogous Case
  - premise. The precedent case C1 and the current case C2 have factors in common which favor party P.
  - premise. C1 was decided in favor of party P.
  - conclusion. C2 should be decided in favor of party P.
- Distinguish Analogous Case (Example of an Undercutter)
  - premise. F, a factor favoring P in the precedent case C1, is not in the current case C2.
  - premise. C1 was decided in favor of party P
  - conclusion. The precedent case C1 does not apply to the current case C2

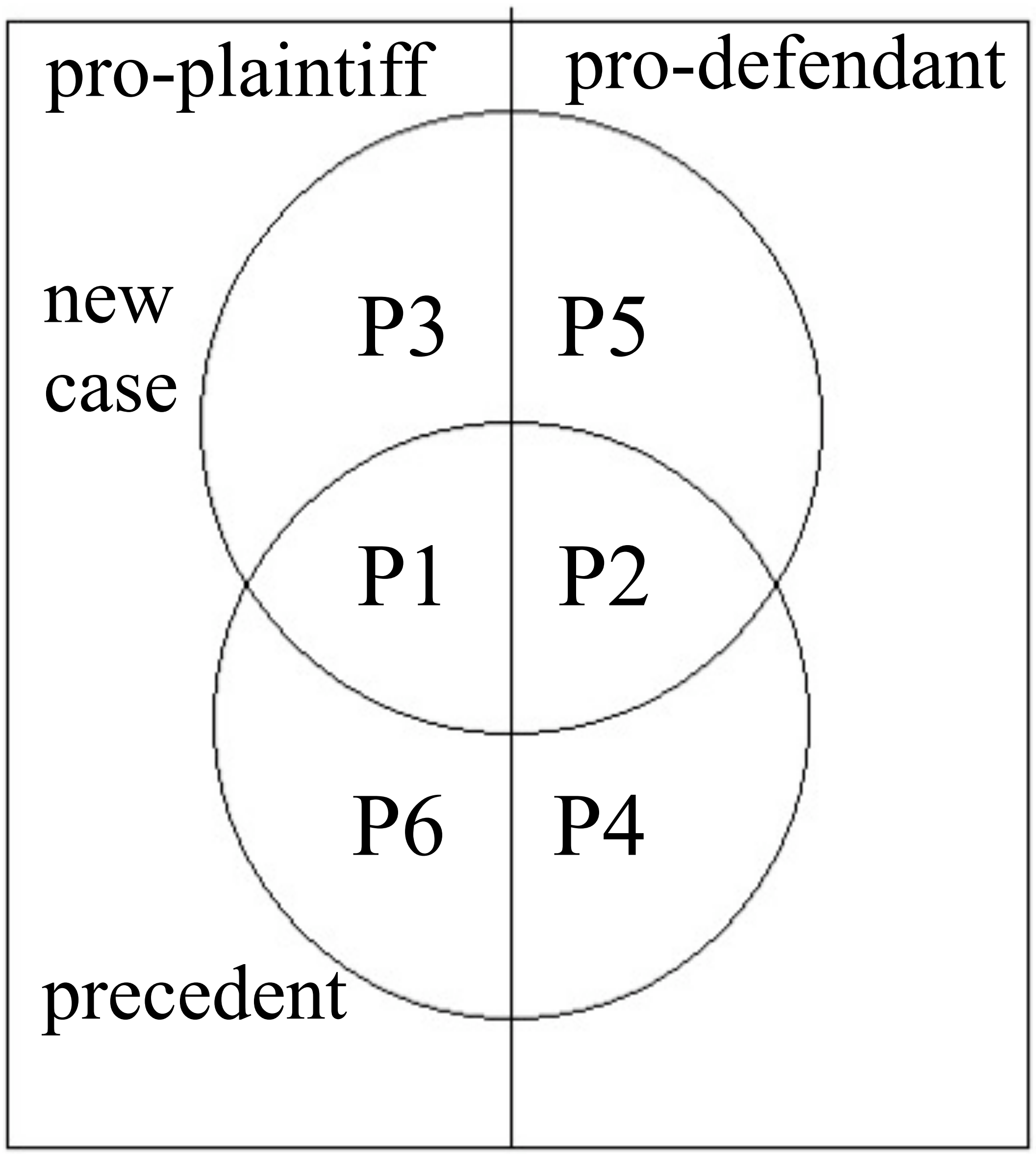
HYPO Preference Order on Arguments – On Pointedness

- A precedent C1 is more "on point" than a precedent case C2 if and only if C1 has more factors in common with the current case than C2

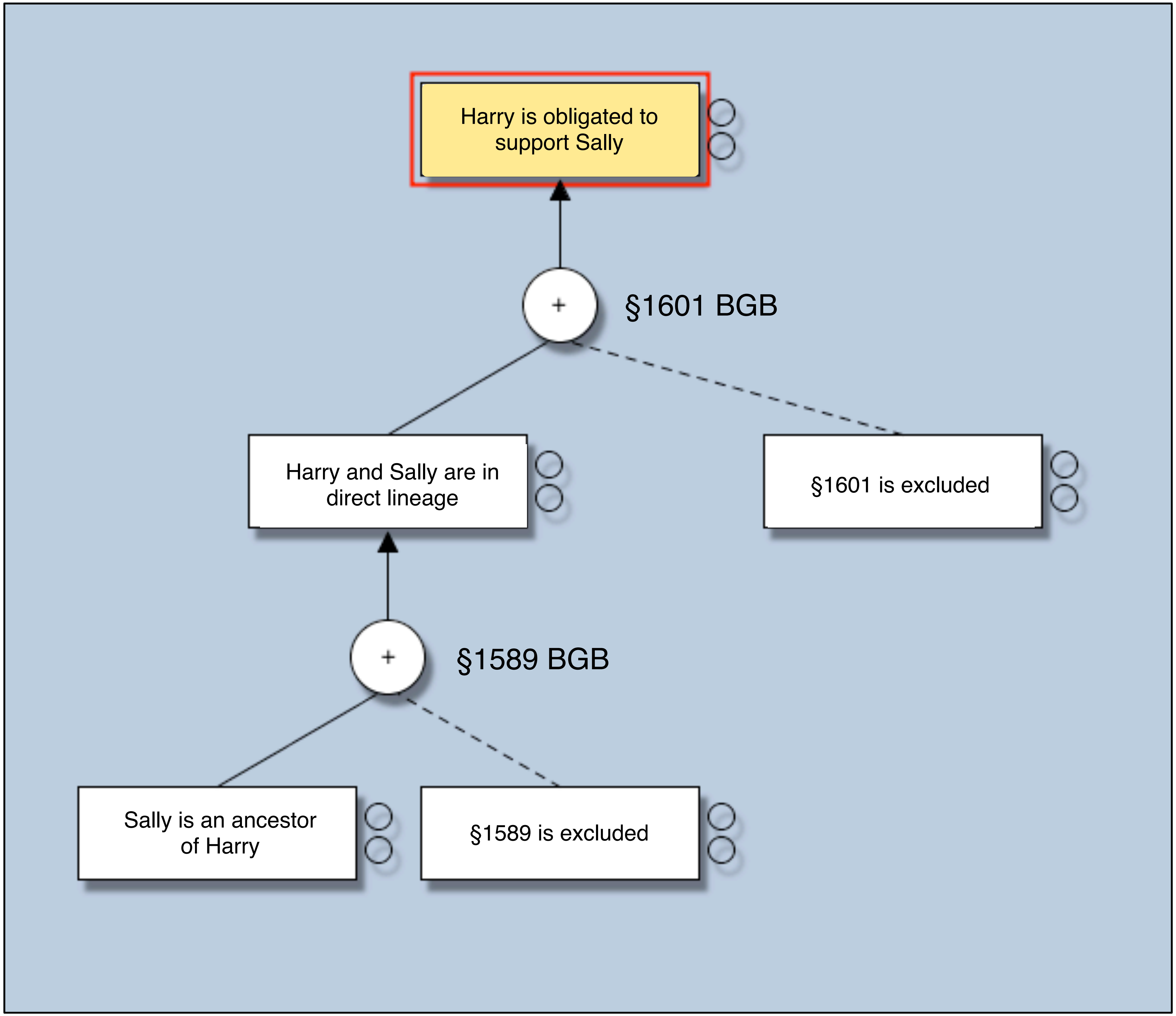
- Let F1 be the factors of C1
  F2 be the factors of C2 and
  F3 be the factors of the current case.

- Then C1 is more on point than C2 iff $|F1 \cap F3| > |F2 \cap F3|$.

Using Partitions to Analogize, Distinguish and Downplay



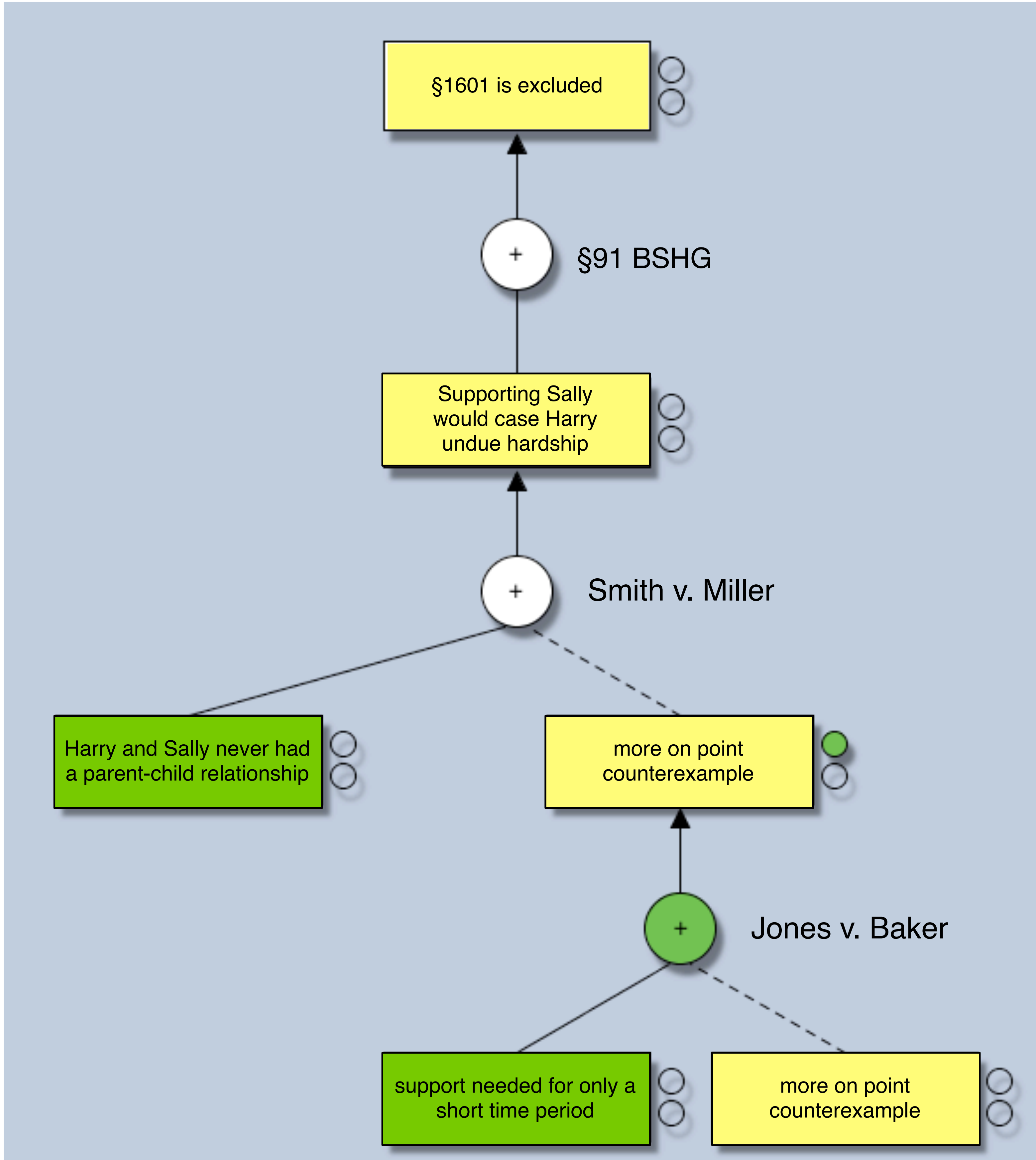
- P1 and P2 factors are used to match cases and argue by **analogy**.
- P5 and P6 factors are used to **distinguish** the PC from the CC, and weaken the argument by analogy.
- P3 and P4 factors are used to **downplay** distinctions based on P5 and P6 factors
- Source: Wyner & Bench-Capon, 2007


Harry is obligated to support Sally
§1601 BGB
Harry and Sally are in direct lineage
§1601 is excluded
§1589 BGB
Sally is an ancestor of Harry
§1589 is excluded

Undue Hardship Rule

```
rule s91-BSHG.
    s1601-BGB excludes “Person1 is obligated to support Person2”
given
    “Person1 is obligated to support Person2” would cause Person1 undue hardship
```

But “undue hardship” is undefined in the statute and left open for the courts to interpret.


§1601 is excluded
§91 BSHG
Supporting Sally would case Harry undue hardship
Smith v. Miller
Harry and Sally never had a parent-child relationship
more on point counterexample
Jones v. Baker
support needed for only a short time period
more on point counterexample

On the Need for “Bridging Rules” [Prakken 2008]

- The example above has been simplified.
- The premise of the argument from §91 BSHG is a sentence in predicate logic. More formally:

```
would-cause-undue-hardship(obligation-to-support(Harry, Sally)), Harry)
```

- But the HYPO/CATO style of case-based reasoning is propositional (“factors”).
- Thus we need “bridge rules” mapping first-order formulas to propositional letters to combine arguments from rules with arguments from cases.
- To combine arguments from other argumentation schemes, additional bridge rules may be needed.

The Carneades Implementation of these Schemes

- Argument from Rules
  - Logic programming technology rule engine
  - Extended to construct arguments and support dialectical negation, exceptions, and argumentation about rule priorities.
- Argument from Ontologies
  - Supports the DLP dialect of Description Logic
  - Ontologies can be represented using KRSS syntax or imported from OWL
- Argument from Cases
  - Implements Bench-Capon and Wyner's reconstruction of Aleven's CATO system.
- Details
  - Open Source: http://carneades.berlios.de
  - Written in R6RS Scheme

Main Points

- Two roles of argumentation schemes: patterns and methods
- Our original contribution is to a computational model of some legal argumentation schemes, in their role as argument generators, which enables arguments from hybrid forms of legal reasoning to be integrated
- The Carneades system implementing this model is designed to be an interactive **argumentation assistant**, not a fully automated reasoner.

Acknowledgments

- This work took place within the ESTRELLA European project (IST-4-027655), in cooperation with the partners in the project, including
  - Alexander Boer, Trevor Bench-Capon, Joost Breuker, Giuseppe Contissa, Tom van Engers, Rinke Hoekstra, Monica Palmirani, Giovanni Sartor, Radboud Winkels and Adam Wyner.
- We would also like to thank Henry Prakken for his collaboration with us on Carneades and fruitful discussions about combining forms of reasoning using bridge rules.
- An earlier version of this talk was presented at the CMNA Workshop in 2008.