# Survey of Knowledge Representations for Rules and Ontologies

Benjamin Grosof*

October 24, 2013

Ontolog Forum‡

Globally accessible webconference session

* Benjamin Grosof & Associates, http://www.mit.edu/~bgrosof/

and

Coherent Knowledge Systems http://www.coherentknowledge.com

‡ http://ontolog.cim3.net/cgi-bin/wiki.pl?ConferenceCall_2013_10_24

# *Concept of KR*

- KR = Knowledge Representation
- A KR S is defined as a triple (LA, LC, |=), where:
  - LA is a formal language for specifying sets of assertion statements
  - LC is a formal language for specifying sets of conclusion statements
    - LC is not necessarily even a subset of LA. E.g., in declarative logic programs (LP). In first-order logic (FOL), LC is the same as LA.
  - |= is the <u>entailment</u> relation.
    - A |= C means C is sanctioned as a conclusion from the set of assertions A.
    - Conc(A,S) stands for the set of conclusions that are entailed by A in KR S. We assume here that Conc is a function.
    - Typically, e.g., in FOL and LP, entailment is defined formally in terms of <u>models</u>, i.e., truth assignments that satisfy the premises and meet other criteria.

# ***Background: Example KR's***

1. Relational databases: relational algebra, cf. SQL
   - A sub-kind of declarative Logic Programs (function-free Horn)
2. Mathematical classical logic: first-order logic (FOL), higher-order logic. *Most people learn it in school.*
   - E.g., used in program verification, and planning.
3. Rules in various flavors
   - Central abstraction: declarative Logic Programs (LP)
     - *Most people do NOT learn LP in school*
   - Key extension: Rulelog
4. Many others:
   - Bayesian probabilistic networks, Probabilistic LP, Markov Logic Networks, fuzzy logic; inductive, possibilistic, …
   - Modal logics, description logics, temporal logics, …
   - Answer Set Programs (another extension of LP)

# *What are “Ontology” and “Rule”, in general*

- Ontology is a purpose/subset of knowledge:  definitional in flavor
  - A key aspect is:  terminology
  - *Ex.:  Lions are a subcategory within felines*
  - *Ex.:  Every health care visit has a required copayment amount*
- Rule is an if-then logical implication.  A fact is a special case of a rule.
  - *Ex.:  During the mitosis phase of an animal’s cell cycle, all DNA is replicated*
  - *Ex.:  AAA members get a weekend discount of 20% on suites, at hotel chain X*
- Almost any kind of rules – or other logical – knowledge can be viewed as consisting of definitions  … and thus “ontological” in a sense
  - Necessary and sufficient conditions for when a concept/relation/expression is true/false.  E.g., cf. “concept learning” in empirical induction.
- “Rules” and “ontologies” are overlapping, not disjoint!  (in general)
- Some KR’s are aimed at particular kinds of ontological knowledge
  - E.g., Description Logic
  - As shorthand, knowledge specified in such a KR is called an “ontology”
    - Yet much of this knowledge may be facts rather than definitions.
    - (This can lead to confusion.)

# *Some Common Kinds of Ontological Knowledge*

- Two common kinds of ontological knowledge are:
  - <u>Formalized vocabulary</u>
  - <u>Schemas</u>, e.g., of databases or object-oriented information models
  - These two kinds overlap, in general

- One basic sub-kind of formalized vocabulary is:
  - A list of categories (“classes”): each a predicate of arity 1
  - A subclassof hierarchy among such classes
  - A list of properties (sometimes called “attributes”): each a predicate of arity 2
  - Restrictions on the domain and range of each property
  - (Anti-) reflexivity, symmetry, and/or transitivity of various properties
  - (Non-) disjointness or equivalence of various pairs of classes or properties

- Description Logic: aimed at ontological knowledge
  - The KR basis for OWL and RDF-Schema (which is simpler than OWL)
  - Good for representing: many kinds of formalized vocabularies or schemas; some kinds of categorization/classification and configuration tasks
  - Severely limited in important ways

# *Need for Other Kinds of Ontologies* ***besides OWL***

- Forms of ontologies practically/commercially important in the world today*:
  - SQL DB schemas
  - "Conceptual models" in UML and E-R (Entity-Relationship)
  - OO inheritance hierarchies, procedural interfaces, datatype declarations
  - XML Schema
  - OWL is still emerging, wrt deployed usage – dwarfed by all the above
  - RIF – early emerging
  - LP/FOL/BRMS predicate/function signatures
  - Built-ins (e.g., SWRL/RuleML)
  - Equations and conversion-mapping functions
- Overall relationship of OWL to the others is as yet largely unclear
  - There are efforts on some aspects, incl. ODM (bridge to UML).
  - Bright spot is OWL-RL relationship to RIF: formulated as a set of RIF-BLD axioms.
- OWL cannot represent the nonmonotonic aspects of OO inheritance
- OWL does not yet represent, except quite awkwardly:
  - n-ary relations
  - ordering (sequencing) aspects of XML Schema

- (*NB: Omitted here are statistically flavored ontologies that result from inductive learning and/or natural language analysis.)

# Declarative Logic Programs (LP) is the Core KR today

- **LP is the core KR of structured knowledge management today**
  - **Databases**
    - Relational, semi-structured, RDF, XML, object-oriented
    - SQL, SPARQL, XQuery
    - Each fact, query, and view is essentially a rule
  - **Business Rules – the commercially dominant kinds** (see next slide)
  - **Semantic Rules**
    - RuleML standards design, incl. SWRL. The main basis for RIF.
    - W3C Rule Interchange Format (RIF): -BLD, -Core. E.g., Jena tool.
  - **Extension: Rulelog.** E.g., Coherent's tool.
  - **Semantic Ontologies**
    - W3C RDF(S)
    - W3C OWL-RL (= the Rules subset). E.g., Oracle's tool for OWL.
  - **Overall: LP is "the 99%", classical logic is "the 1%"**

- **Relational DB's were the first successful semantic technology**
  - LP is the KR/logic that was invented to formalize them
- **The Semantic Web today is mainly based on LP KR** … and thus essentially equivalent to semantic rules
  - **You might not have realized that!**

# *Commercially Dominant Legacy Kinds of Business Rules*

- E.g., in OO applications, workflows

- Production rules (OPS5 heritage): e.g.,
  - IBM ILOG, Fair Isaac, Drools, Oracle, Jess: rule-based Java/C++ objects.
- Event-Condition-Action (ECA) rules (loose family), cf.:
  - business process automation / workflow tools.
  - active databases; publish-subscribe.
- Prolog. *"logic programs": as a full programming language*
  - *"Logic programming" is different from "declarative logic programs"*

- LP is the core KR for production rules, ECA rules, and Prolog
  - … insofar as they are semantic (i.e., "declarative")
  - But they are each only partially semantic

# *KR View of Semantic Web related standards*

***Hazy wrt Standardization: more Framework, incl. about:***

***– Uncertainty (probabilistic, fuzzy); Provenance (proof, trust)***

**Logical Framework standards/designs: RIF-FLD, RuleML**

## LP (Logic Programs) family

- **Umbrella standards/designs**
  - RIF-Rulelog
  - RuleML-LP
- **Database Query Standards***
  - SQL
  - SPARQL
  - XQuery
- **Business Rules Families***
  - Production
    - RIF-PRD
  - ECA (Event-Condition-Action)
  - Prolog

## Classical Logic

- **Umbrella standards/designs:**
  - CL (ISO Common Logic)
  - RuleML-FOL
- **Semantic/Web Standards (other)**
  - RDF
  - RDFS (Schema)
  - OWL RL (Rule Profile)
  - RIF-BLD (Basic Logic Dialect)
    - (and SWRL)
  - OWL DL (Description Logic)
  - OWL Full
  - SBVR (OMG Semantic Business Vocabulary and Rules)

*Via KR mapping to LP, maybe with restrictions

# KR View of Semantic Web related standards

***Hazy wrt Standardization: more Framework***

***– Uncertainty (probabilistic, fuzzy); Provenance (proof, trust)***

**Logical Framework standards/designs: RIF-FLD, RuleML**

## LP

- **Horn**
- **Rest**

## Classical Logic

- **Umbrella standards/designs:**
  - CL (ISO Common Logic)
  - RuleML-FOL
- **Semantic/Web Standards (other)**
  - RDF
  - RDFS (Schema)
  - OWL RL (Rule Profile)
  - RIF-BLD (Basic Logic Dialect)
    - (and SWRL)
  - OWL DL (Description Logic)
  - OWL Full
  - SBVR (OMG Semantic Business Vocabulary and Rules)

*

*Via KR mapping to LP (sound, nearly complete)

# *Classical Logic – Family of KR's*

- Fully general case: Higher-order logic (HOL) – used foundationally in mathematics
  - A predicate or function itself is a ***term*** (e.g., a variable, or even a complex term)
  - Its generality makes it very difficult to automate fully
  - As usual:
    - An atom is a predicate applied to an argument tuple of terms
    - A term is a constant, a (logical) variable, or a complex term
    - A complex term is a (logical) function applied to an argument tuple of terms
    - Formulas are formed from atoms by applying:
      - Quantifiers: $\forall$, $\exists$
      - Connectives: $\neg$, $\wedge$, $\vee$, $\Leftarrow$, $\Rightarrow$, $\Leftrightarrow$

- First-order logic (FOL) – used in computer science much more than is HOL
  - Restriction: each predicate or (logical) function must be a ***constant***
  - Much more amenable to automation than higher-order
  - Used in program verification, planning/scheduling constraint satisfaction

- Description Logic (DL) – used for ontologies in OWL. Actually, a sub-family.
  - Restricts patterns of variable appearances in certain ways
  - First-order. No functions.

# *Declarative* *Logic Programs (LP) – Family of KR's*

- Normal LP
  - Rule syntax: $H \leftarrow B_1 \wedge \ \ldots \wedge B_k \wedge \text{naf } B_{k+1} \wedge \ldots \wedge \text{naf } B_m$ . $(m \geq 0)$
    - H and Bi's are atoms.
    - $\leftarrow$ is a kind of implication that lacks contraposition.
      Its lhs and rhs are called the rule's "head" and "body", respectively.
    - naf ("negation-as-failure") is a kind of negation that is logically non-monotonic. Intuitively, naf Bi means "not believe Bi".
  - Semantics (well-founded) is defined constructively via an iterated fixed point.
    - It has 3 truth values: *true*; *false* in the naf sense; and an intermediate "*undefined*", which can represent paradoxicality.

- Rulelog: extends normal LP. Adds several expressive features:
  - Meta knowledge – several aspects
    - Hilog (see next slide). Reification: formula can be treated as a term.
    - Defeasibility: rules can have exceptions, behaving non-monotonically
    - Rule id's: enables meta-statements about assertions, incl. for provenance
    - Restraint: bounded rationality, using the "undefined" truth value
  - Omniformity: classical-looking formulas can appear in head and body
  - *See Ontolog Forum 2013-06-20 session presentation for details.*

# *Important Restrictions* ***(NB: can be combined)***

- ***Each of the restrictions below applies not only to Classical Logic but also to Logic Programs, Rulelog, and many other KR's***

- Hilog – important extension of first-order
  - Syntax is higher-order (a bit restricted)
  - Semantics reduces to first-order, however (via transformation)
  - Used in Common Logic (ISO), and thus
  - Used in Rulelog (draft RuleML/W3C standard)

- First-order
  - Each predicate or (logical) function is a *constant*

- Horn: every formula is a clause in which at most 1 literal is positive
  - Used in databases (SQL, SPARQL, XQuery), RIF-BLD, RDF(S)
  - Point of departure for normal LP and OWL-RL

- Function-free: no functions
  - Used in databases (SQL, SPARQL, Xquery), RIF-Core, OWL, RDF(S)

- Propositional: arity is zero. This is a further restriction of function-free.
  - Used in constraint satisfaction

# *Summary of Computational Complexity of KR's*

- *For task of inferencing, i.e., answering a given query.*
  - *Tractable = time is polynomial in n, worst-case; n = |assertions|*
  - *Also: m = # of atoms ($m \leq n$). v = max # of distinct variables per rule.*

- FOL propositional: co-NP-complete, i.e., "exponential"
  - Blowup due to "reasoning-by-cases" with disjunctions
- FOL: undecidable
  - Blowup due to recursion thru functions

- Horn LP propositional: O(n), i.e., linear
- Normal LP propositional: $O(n \cdot m)$, i.e., quadratic
- Normal LP function-free: polynomial, if v is a constant (as is typical in practice)
- Horn or Normal LP: undecidable
  - Blowup due to recursion thru functions

- Rulelog: polynomial, if one employs the restraint feature (as is typical in practice)
  - With functions – and other features (hilog, defeasibility, etc.) that extend LP
  - Leverage "undefined" truth value to represent "not bothering"

# *Relationships/Bridges Between Classical and LP Families of KR*

- Fundamental Theorem connects Horn LP to Horn FOL
  - Horn LP entails the same set of ground atoms as Horn FOL
    - (when $\leftarrow$ is replaced by $\Leftarrow$ )
  - Horn LP is sound but incomplete wrt Horn FOL, which has additional non-ground-atom conclusions, notably: non-unit derived clauses; tautologies
- OWL-RL practical reasoning is thus essentially LP. Ditto RDF(S).

- Generalization: Rulelog is sound but incomplete wrt hilog FOL
  - (Certain restrictions apply)
  - Rulelog lacks "reasoning-by-cases"
    - Essentially it has the power of the unit resolution proof strategy
  - Rulelog reasoning in presence of conflict is usefully selective unlike hilog FOL
    - Rulelog has the defeasibility feature, i.e., handles conflict … while retaining a consistent set of conclusions
    - By contrast, classical logic is perfectly brittle: any conflict results in all sentences being concluded (i.e., garbage)

# *KR View of Semantic Web related standards*

***Hazy wrt Standardization: more Framework, incl. about:***

***– Uncertainty (probabilistic, fuzzy); Provenance (proof, trust)***

## Logical Framework standards/designs: RIF-FLD, RuleML

### LP (Logic Programs) family

- **Umbrella standards/designs**
  - RIF-Rulelog
  - RuleML-LP
- **Database Query Standards***
  - SQL
  - SPARQL
  - XQuery
- **Business Rules Families***
  - Production
    - RIF-PRD
  - ECA (Event-Condition-Action)
  - Prolog

### Classical Logic

- **Umbrella standards/designs:**
  - CL (ISO Common Logic)
  - RuleML-FOL
- **Semantic/Web Standards (other)**
  - RDF
  - RDFS (Schema)
  - OWL RL (Rule Profile)
  - RIF-BLD (Basic Logic Dialect)
    - (and SWRL)
  - OWL DL (Description Logic)
  - OWL Full
  - SBVR (OMG Semantic Business Vocabulary and Rules)

*Via KR mapping to LP, maybe with restrictions

# *KR View of Semantic Web related standards*

***Hazy wrt Standardization: more Framework***

***– Uncertainty (probabilistic, fuzzy); Provenance (proof, trust)***

**Logical Framework standards/designs: RIF-FLD, RuleML**

## LP

- **Horn**
- **Rest**

## Classical Logic

- **Umbrella standards/designs:**
  - CL (ISO Common Logic)
  - RuleML-FOL
- **Semantic/Web Standards (other)** *
  - RDF
  - RDFS (Schema)
  - OWL RL (Rule Profile)
  - RIF-BLD (Basic Logic Dialect)
    - (and SWRL)
  - OWL DL (Description Logic)
  - OWL Full
  - SBVR (OMG Semantic Business Vocabulary and Rules)

*Via KR mapping to LP (sound, nearly complete)

# *For More Info*

See the ff. longer AAAI-13 Rules tutorial, available at http://coherentknowledge.com/publications :

Benjamin Grosof, Michael Kifer, and Mike Dean. Semantic Web Rules: Fundamentals, Applications, and Standards (abstract). Conference Tutorial (Slides for 4-hour tutorial), 27th AAAI Conference on Artificial Intelligence (AAAI-13), Bellevue, Washington, July 15, 2013.

This is the latest iteration of a tutorial that since 2004 has been presented at numerous scientific conferences on web, semantic web, and AI.

A book is in early stages of preparation based on this tutorial.

# Acknowledgements

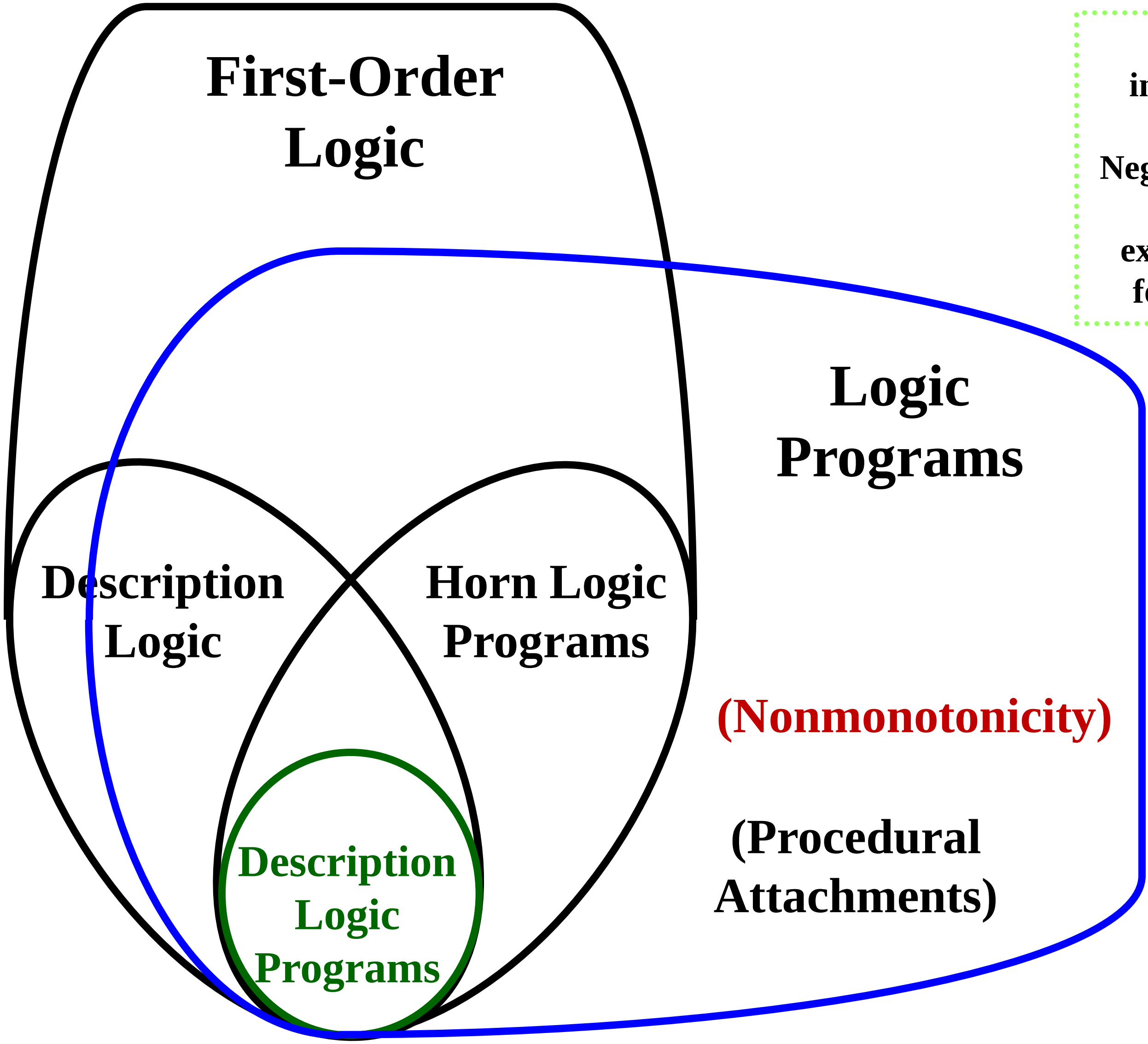
- Thanks to Michael Kifer and Mike Dean, co-authors of longer tutorial presentations upon which this presentation was based.

# Thank You

**Disclaimer: The preceding slides represent the views of the author(s) only.**
**All brands, logos and products are trademarks or registered trademarks of their respective companies and organizations.**

# *OPTIONALS FOLLOW*

# *Venn Diagram: Expressive Overlaps among KRs*




**NB: Nonmon LP, including Courteous, relies on Default Negation as fundamental underlying KR expressive mechanism for nonmonotonicity**

# *The "Spirit" of LP*

The following summarizes the "spirit" of how LP differs from FOL:

- "Avoid Disjunction"
  - Avoid disjunctions of positive literals as expressions
    - In premises, intermediate conclusions, final conclusions
    - (conclude (A or B)) only if ((conclude A) or (conclude B))
  - Permitting such disjunctions creates exponential blowup
    - In propositional FOL: 3-SAT is NP-hard
    - In the leading proposed approaches that expressively add disjunction to LP with negation, e.g., propositional Answer Set Programs
  - No "reasoning by cases", therefore
- "Stay Grounded"
  - Avoid (irreducibly) non-ground conclusions

LP, unlike FOL, is straightforwardly extensible, therefore, to:

- Nonmonotonicity – defaults, incl. NAF
- Procedural attachments, esp. external actions

# *Examples – slide TODO ideally*

- Higher-Order not First-Order
- First-Order Non-Horn
- Horn First-Order

- For now, see the AAAI-13 rules tutorial