Some Analysis of KRR Requirements and Directions for Neuro-Symbolic Decision Support

Benjamin Grosof

DARPA
*** based on research done while at Coherent Knowledge**

https://www.darpa.mil/staff/dr-benjamin-grosof
http://benjamingrosof.com
http://www.linkedin/com/in/benjamingrosof
https://coherentknowledge.com

Invited Talk (30 min.) at AAAI-2024 NucLeaR Workshop
Feb. 26, 2024
https://neurosymbolic.github.io/nsss2023/

Acknowledgements and Thanks: This presentation includes, in part, work done in collaboration with: Michael Kifer, Alex Gray, Janine Bloomfield, Theresa Swift, and Paul Fodor.

Version of Feb. 26, 2024,
(Revised from earlier versions.)

Preface

In this talk, I am NOT speaking for DARPA
- the content reports research work done before joining DARPA
- forall ?X in talk content { DARPA does not endorse ?X }
- eg DARPA does not endorse desiderata, LP, Rulelog, ErgoAI, XSB, Problog2, PITA, etc.

Apologies for not enough graphics/illustrations (due to low prep time)

For a deeper dive on the KRR technically, see the 2-hour "KRR Tutorial for Neuro-Symbolic Software:
Concepts, Desiderata, Logic Programs, Expressiveness" presented at the Neuro-Symbolic AI Summer School held Aug. 29-30, 2023;
- home: https://neurosymbolic.github.io/nsss2023/
- additional content, incl. talk videos, at:
    - o https://twitter.com/asimunawar/status/1696850589299229130

Focal Goals for Neuro-Symbolic AI

- Tighter/finer combination/composition of
  induction + deduction
- Harness:
  human <u>k</u>nowledge
  + principled <u>ded</u>uctive reasoning,
  then learn starting from that K+ded
  - "Evolution's lesson"
  - Sample efficiency
- Support well: Social process of iterative system development
  - Transparency, explainability, interpretability for
    high accuracy, generalization, trustworthiness

Induction (ML) Requires Deduction (KRR)

- How most ML induction (“training”) works today:
- Iteratively search for models
  (typically with some stochasticity, stopping when returns diminish)
  - Analyze accuracy of **predictions** on focal examples
    (typically via error distance penalties, often on subsamples)
    - Involves a heavy amount of **inner-loop** **deduction**
  - Analyze simplicity of model
    (typically: penalty for more parameters or other complexity of model)
  - Analyze goodness of agreement with other previous knowledge
    (typically via penalty for distance from previous knowledge)
    - Involves a heavy amount of **inner-loop** **deduction**
  - Feed back the analysis above to directions for the next iteration
    (typically: differential gradients)
- In NN, the deductive KR has low logical expressiveness
- Some other ML approaches have more expressive KR

Need Scalability of Deduction

- Deduction gets hit hard in most ML
  - Search lots of inductive hypotheses
- ➔ requirements for NS deduction:
  - Computational scalability: often, at least tractability
  - Often: completeness is needed for soundness of defeasible ded
- Decision support typically requires reasoning about action
  - E.g., planning, ethical, legal, biz policies, contracts, financial, science, health treatment, military orders & regs, autonomous vehicles/cyber-physical, games
  - ➔ need defeasibility (e.g., for causal effects/persistence)

Practical Logic, vs. Classical Logic

- Goal: support IT, vs. mathematics
  - E.g., Databases, Knowledge Graphs, Rules
  - Central: declarative logic programs (LP) KR. Non-classical.
    - Essentially: an extension of the logic of databases.
    - **LP is the core KR of structured knowledge management today**
- Requirements:
  - Scalable computationally … and sociotechnically; →→
    - Reliable. Explainable. Expressive/meta. Compositional intra-/inter-. Friendly to: structured info, natural language, software/engineering.
  - Robust in face of human errors and miscommunications; →→
    - "Humble", i.e.:
      - Avoid general proof by contradiction
      - Avoid general reasoning-by-cases
        - *Background: What is "reasoning-by-cases":*

          Assertions: if A then C. if B then C. A <u>or</u> B.

          Conclude: C.

          It's at heart of 3-SAT, the original NP-hard problem.

Industry Landscape of Practical Logic

- ***LP is the core KR of structured knowledge management today***
- Subsets of LP:
  - Relational databases (SQL) *[function-free is a key restriction]*
  - Knowledge graphs (KGs), a.k.a. graph databases (SPARQL) *[similar subset]*
  - Production rules, Event-Condition-Action rules. More precisely: their logical subsets.
  - Prolog. More precisely: its logical subset (a.k.a. “pure”). E.g., ATS huge application at US CBP.
  - Many RuleML & W3C RIF dialects, e.g., RIF-BLD, RIF-Core, SWRL *[NAF-free]*
  - Many ontology standards, e.g., OWL-RL (& RDF-Schema) *[similar to KG]*
- Other:
  - Subsets of Classical Logic:
    - Propositional. E.g., for hardware circuit design, satisfiability for planning.
    - First Order Logic (Common Logic). E.g., for program verification.
  - Various niche or not so commercially/practically important
    - Temporal. Modal. Description Logic subset of FOL.
  - Emerging:
    - Rulelog (RIF-Rulelog dialect in draft) – combines several major LP extensions
      - Covers: most RuleML & RIF dialects; Probabilistic LP
    - Probabilistic LP cluster (Problog, PITA, …). Bayesian, Fuzzy / Prob. Soft Logic.
    - Others not so commercially/practically prominent as yet
      - Answer Set Programs, MKNF. Related to LP & Rulelog, but closer to classical, less humble.

KRR Desiderata and Foci (I)

- *Suitability for application set.* Consider: decisions, knowledge sources, analytics.

- *Expressiveness*: flexibility about what+how you can state, edit, and infer
  - Syntax matters, sociotechnically. Expressiveness ≠ (ir)reducibility. It's about what one can *state+edit+infer conveniently/concisely+clearly*.
  - Several expressive/extension features are highly desirable for various kinds of knowledge and application usages. Creativity: restrictions+extensions

- *Scalability computationally*: preferably, practical tractability, i.e., polynomial time

- *Scalability sociotechnically*: this has many aspects

KRR Desiderata and Foci (II)

- *Compositionality* (including: modularity, reusability, ease of updating/modifying)
  - In knowledge (assertions & conclusions)
    - Ex.: *Locales* in LP/Rulelog: set of rules that mention a literal. Can walk syntax dependencies. Proof procedures, including incremental, exploit it.
  - As software component in overall system engineering.
    - Orchestration via querying & data transfer: outbound & inbound.
    - Convenient interfaces, for typical enterprise data & environments
  - Standardization for interoperability, optimizations

- *Explainability* (including: interpretability, transparency)
  - Comprehensibility, including by non-programmers, non-AI-experts
  - Crucial for debuggability, accuracy, trustability

- *Robustness* in face of contradiction-type conflict, avoiding brittleness of classical

- *Completeness*: is *often required* for even soundness (e.g., in defeasibility)

Some More Key Expressive Features

- Defeasibility
  - NAF, strong negation, argumentation, priorities, rule id's, annotation
- Higher-order syntax (HiLog) and reification
  - e.g., for multi-agent (nestedly) belief/desire/intention and deontics
- Probabilistic uncertainty
  - Bayesian, upper-lower bounds, independence/correlation
  - fuzzy/T-norm, lattice, other
- Restraint bounded rationality
- Strong Meta overall

Some State-of-the-Art LP KRR Tool Systems relevant to NS

NB: All these are open source.

- ELP: XSB (T. Swift, D. Warren, et al)
  - Full programming language that is Prolog++
  - Pioneered well-founded semantics & "tabling" techniques. Good soundness & completeness.
- ELP: SWI Prolog (J. Wielemaker et al)
  - Most popular Prolog/Prolog++, but less expressive and complete than XSB.
  - Collaborates and swaps code with XSB. E.g.:
  - Since v8.2 (2019): Brought (much) advanced tabling support of XSB to SWI-Prolog, for: well founded semantics, restraint, incremental tabling, and shared tabling
- Janus: Python-XSB interface: bi-directional (T. Swift)
  - Speedy & easy-to-use.
  - Combine tightly: Python+Prolog. Optionally: over C, and under ErgoAI.
- ELP – Rulelog: ErgoAI (Coherent Knowledge; B. Grosof, M. Kifer, T. Swift, et al)
  - Full programming language that is XSB++ thus Prolog+++
  - Tightly combines several major expressive extensions: higher-order syntax, defeasibility, restraint, …
- ELP – Probabilistic LP: Problog 2 (L. de Raedt et al). Bayesian; *has ML.*
- ELP – Probabilistic LP: PITA/cplint/PLOW (T. Swift, F. Riguzzi, B. Grosof).
  - Non-Bayesian & Bayesian; cplint *has ML.*

Why LP? w.r.t. *NSS Desiderata* (I)

- *Scalability*
  - Computationally – volume and velocity.
    - Inferencing is tractable for very broad cases. Far superior to classical.
  - "Socially" – variety of knowledge sources and contexts
    - Strong semantics, interoperability, standards, practically supported
    - Defeasibility (robust conflict handling). Far superior to classical.
- *Compositionality*
  - High. Fundamentally, due to: semantics, proof theory & algorithms.
  - "Locale" = set of rules that mention an atom. Modules (info hiding). Orchestration: external querying, reactivity, built-ins, constraints.
- *Explainability* (requires symbolic)
  - Based on symbolic, logic. Natural deduction style of proof; intuitionistic.
- *Reliable reasoning, principled uncertainty of conclusions*
  - Strong semantics, including extensions for higher-order, defeasibility, restraint, probabilistic. Far superior to classical.

Elaborations about "Spirit" of LP

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

- "Avoid Head Disjunction"
  - Avoid head disjunctions of positive literals as expressions
    - In assertions, 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 (unlimited) "reasoning by cases", therefore
- "Stay Grounded"
  - Avoid (irreducibly) non-ground conclusions, i.e., be ground-able

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

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

Some Open Challenges in KRR for NS

- Natural Language mapping of text to logic
- **Tractable** probabilistic in combination with other expressiveness
  - Combining tightly many KRR expressive features in one overall KRR
    – Motivations: flexibility; interoperation "umbrella"
  - Probabilistic as an expressive feature of KRR is a key potential/required bridge between ML and KRR
    – But: unrestricted Bayesian probabilistic is intractable computationally, when combined with LP or other medium-to-high logical KRR expressiveness overall
- Overall: learning-friendly **+** human-cognition-friendly

KRR Desiderata and Foci (III)

- *Tradeoffs, tradeoffs, tradeoffs!!!* → Seek Pareto-optimal sweet spots.
- *Rulelog ELP is a sweet spot:* tightly combines several major expressive features, plus: good soundness & completeness, strong explainability, strong modularity & orchestration, robustness, attractive computational scalability (polynomial), many optimizations, commercial-quality engine implementation (ErgoAI); open-source
  - Transformational stack: ErgoAI over XSB (Prolog+) over C
  - Janus: Python-Prolog speedy easy-to-use bidirectional (over C)
    - Compose {XSB, ErgoAI} with: ML, NLP, other components/services
  - Current work: tight integration of Probabilistic feature, and ML: via PITA/cplint/PLOW
- *Probabilistic LP cluster of ELP is a sweet spot:* includes{Problog, PITA, others}
  - Problog2: implemented in Python; open-source; ML tool(s) too
  - PITA: over {XSB | SWI} (Prolog+); ML tool too: cplint; open-source
  - Fabrizio Riguzzi book Foundations of PLP, 2nd ed. (2022): is good source of research knowledge
    - Also: 2018 survey paper by Riguzzi & Teri Swift about the PLP cluster
  - PLOW is an extension of PITA, cross-fertilizing with Rulelog
    - Combines Bayesian & generalized fuzzy/T-norms, via: intervals, well-founded, restraint, defeasibility; also other uncertainty & weighting
    - [Paper & initial prototype 2019. + current work, by Grosof, Swift, & Riguzzi. Open-source.]

Conclusions: KRR Directions for NS

- Aim to harness human K+ded, incl. for sample efficiency
- (Extended) LP family beyond knowledge graphs should be a major focus, e.g., as point of departure
  - Well-founded flavor for sake of scalability
  - Rulelog; Problog2, PITA are Pareto-optimal expressively
- Open challenges in KRR: text-to-logic, tractable probabilistic, combining features
- Overall, aim for new fundamental
  rigorous concepts, theory, algorithms for NS
  that better meet NS desiderata
  by incorporating KRR insights and SOTA approaches

Thank You