Radial Restraint: A Semantically Clean Approach to Bounded Rationality for Logic Programs†

Benjamin Grosof¹,³ and Terrance Swift²,³

Paper presentation (15-min.) and also poster presentation
27th AAAI Conference on Artificial Intelligence (AAAI-13)‡
July 17, 2013, Bellevue, Washington, USA


† Work partly supported by Vulcan, Inc., http://www.vulcan.com
Overview

• Motivation: extend databases to richer/meta* knowledge
  • Need logical functions in LP
  • LP = declarative logic programs: the logic of SQL and pure Prolog
• Problem: functions create potential for runaway computation
  • Fundamental in semantics: potentially infinite set of answers to a query
  • Example: num(0). num(succ(?x)) :- num(?x). Query ?- num(?y).
    Answers for ?y: 0, succ(0), succ(succ(0)), succ(succ(succ(0))), …
• New Approach: radial restraint – a form of bounded rationality
  • Specify a bound on normed depth of terms that are reasoned about
  • Extend reasoning procedure (SLG) with term abstraction. Incurs low overhead.
• Leverages “undefined” truth value to represent “not bothering”
  • The other 2 truth values are true and false (cf. negation-as-failure)
  • Extends LP’s well-founded semantics where undefined is used for paradoxicality
  • Sound, even when reasoning is non-monotonic
• Provides “valves” to ensure computational scalability
• Fully semantic, enabling social scalability

* For more info, see [Grosof et al, AAAI-13 Semantic Web Rules Tutorial] – largely about Rulelog
Declarative logic programs (LP) based on the well-founded semantics (WFS) are widely used for knowledge representation (KR).

Logical functions are desirable expressively in KR, but when present make LP inferencing become undecidable.

In this paper, we present radial restraint: a novel approach to bounded rationality in LP.

Radial restraint is parameterized by a norm that measures the syntactic complexity of a term, along with an abstraction function based on that norm.

When a term exceeds a bound for the norm, the term is assigned the WFS's third truth-value of undefined.

If the norm is finitary, radial restraint guarantees finiteness of models and decidability of inferencing, even when logical functions are present.

It further guarantees soundness, even when non-monotonicity is present.

We give a fixed-point semantics for radially restrained well-founded models which soundly approximate well-founded models.

We also show how to perform correct inferencing relative to such models, via SLGABS, an extension of tabled SLG resolution that uses norm-based abstraction functions.

Finally we discuss how SLGABS is implemented in the engine of XSB Prolog, and scales to knowledge bases with more than $10^8$ rules and facts.
Motivations I

• LP is the logic of databases (SQL, SPARQL, XQuery) and pure Prolog
  • Also it’s the logic of:
    • Production/event-condition-action business rules (the declarative fragment)
      • $ multi-billion industry
    • Rule-based basic semantic web ontologies: RDF-Schema, OWL-RL
  • LP – not FOL – is “the 99%” of practical structured info management today
  • (LP = declarative logic programs)

• The practically-used fragment of LP is primarily function-free
  • All the above except pure Prolog are essentially limited to function-free
  • (Function here means function symbol of arity > 0)
Motivations II

• Want richer expressively, yet also scalable
  • Richness to represent natural language, science, policy (incl. contracts, regulation, law), info integration mappings, meta-knowledge, etc.
    • E.g., cf. Rulelog*, which extends LP
  • Scalability includes:
    • Computational, e.g., tractability (worst-case polynomial time)
    • Social, for authoring/consumption by wide set of organizations/people

• Functions are needed for such higher-abstraction KR/KA
  • Skolemization for existential knowledge
  • Hilog for meta knowledge, e.g., defeasibility and textual logic*
  • (Higher-abstraction means higher level of conceptual abstraction)
  • (KR = Knowledge Representation & reasoning)
  • (KA = Knowledge Acquisition)

* For more info, see [Grosof et al, AAAI-13 Semantic Web Rules Tutorial]
Problem of Runaway

• Problem: functions create potential for runaway computation
  • Due to recursion through functions
  • Fundamental in semantics: potentially infinite set of answers to a query
  • Leads to undecidability of inferencing

• Example: numberline
  • Assert
    • number(0).
    • number((s(?x)) :- number(?x).  // s is a function, mnemonic for “successor”
                                         // “?” prefixes a variable. “:-” can be read as “if”.
  • Query   ?- num(?y).
  • Set of answers is infinite: 0, s(0), s(s(0)), s(s(s(0))), …

• FOL (first order logic) suffers from this potential for runaway, too
Past Workaround

• Technique often used in practice to stop runaway, e.g., in Prolog:
  • Cut off inferencing when bound on term nestedness depth is exceeded
  • Assign false (naf) truth value beyond that

• Drawbacks
  • Not semantic: One configuration of one instance of one implementation
  • Not sound, when knowledge/reasoning is non-monotonic (e.g., defeasible)
New Approach: Radial Restraint

- A form of bounded rationality
  - Specify a bound on normed depth of terms that are reasoned about
  - Intuition: Provides “valves” to ensure computational scalability
- Leverages “undefined” truth value to represent “not bothering”
  - The other 2 truth values are true and false (cf. negation-as-failure)
  - Extends LP’s well-founded semantics (WFS) which has undefined
- Stops runaway
- Ensures soundness, even when reasoning is non-monotonic
- Fully semantic, enabling social scalability

- Approach includes semantics, extending the WFS
- Previous usage for undefined in WFS is to represent paradoxicality
  - Such paradoxicality can arise from cyclic dependency through naf (a.k.a. “unstratified” naf)
  - Intuition: oscillatory instability is treated as meta-stable
- Example 1: p :- naf p.
- Example 2: p :- naf q. q :- naf p.
Details of Technical Approach

• Develop a technical approach based on term abstraction
• In term abstraction: replace a subterm by a (fresh) variable
• View Herbrand universe/base as a tree of (ground) terms
• Beyond the radial frontier in that tree, everything goes misty, i.e., *undefined*
• Radius bounds a norm on terms
• The norm can be size, nestedness, or more complex
• I.e., one sacrifices (normed) term depth of reasoning beyond what’s anticipated as relevant
• Ensures finiteness, decidability
Example: Numberline, revisited

- Here, radius is 15.

Screenshot is from UI in Vulcan Inc.'s SILK system.
Proof theory and Implementation

• Approach includes full proof theory
  • Extends SLG resolution → SLGABS
  • SLG is state-of-art LP inferencing procedure which uses “tabling”
  • Tabling = query-driven but saves/reuses work on subgoals
    • Mixes backward and forward inferencing

• Implemented in XSB: LP system that supports logical functions and well founded semantics, with excellent completeness
  • Open source

• Scales to knowledge bases with over 100 Million rules and facts

• Incurs low overhead (computationally)
  • Compared to SLG without radial restraint
Follow-on, Ongoing, and Future Work

- Analyze how ensures tractability not just decidability
  - “Dial-your-own” polynomial degree (radius choice is the dial)
- More kinds of restraint: “skipping”, unsafety, unreturn, anytime
  - See [Anderson et al: RuleML-2013 (best demo award); WLPE-2013]
    - About advanced knowledge base debugging in Rulelog
  - Meta power (expressively) of Rulelog enables the bounded rationality to be specified declaratively within the logic/KR (language) itself
- Extend to other logics, incl. classical

- Opens a field – theoretical and empirical – of finding conditions for restraint that are useful
- Perspective: meta power in reasoning → transcend the XOR trade-off between expressiveness versus decidability/tractability in KR

KR = Knowledge Representation
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.