Information-relational semantics of the Fifth system

Anthony Di Franco

01 September 2018

PLP 2018, Ferrara

Overview

Work in progress on purely declarative software system

Uses adaptive evaluation strategy optimizing measures of certainty (cf. PLP 2017)

- Desire to be purely declarative, and
- nature of evaluation strategy

have implications which I am working out.

Project background

More engineering than academic

Origins in professional project: smart contract language for use by small business owners

Informed by my background in control theory and information theory, and

Implementing latent Dirichlet allocation several ways, one slightly novel

Adaptive evaluation recap

Desire to leverage strong optimality results from multi-arm bandit algorithms for reinforcement learning

Make choice of next evaluation step in a goal according to bandit algorithm

Condition on current state of information in the goal

Adaptive evaluation recap

In Prolog, expand goals in body of current goal

- top-down,
- left-to-right

Gives depth-first search.

Minimizes state associated with search, but

Rarely is optimal or even workable for a given problem as-is.

When matching facts reached, propagate them up towards query.

Adaptive evaluation recap

In Fifth, choose next body goal to expand based on bandit algorithm weights.

Yields *best-first search*.

Not just facts, but partial information such as interval constraints can be propagated.

Information can be propagated in any direction, not just towards query.

Cf. Radul and Sussman, Information Propagation Networks.

Work in progress...

Simulation of adaptive evaluation strategy for sample programs implemented.

Currently accumulating enough constraints on how it should work to implement as a full language.

Goal is system with scope of a good probabilistic Prolog:

- Turing complete
- Purely declarative
- Approximately optimal evaluation strategy

Highlights

- **Mechanisms based in clean PL theory for relational**
  - Abstraction
  - Application
  - Composition

…and…

Everything is a relation and everything is measurable.

Abstraction

Based on delimited control operators

Capture a subtree with some named indeterminate variables designated as arguments → predicate definition

- *Mark* the root of the subtree with a name in one operation
- *Capture* from within the subtree in a second operation, designating the variables within the subtree to become arguments

Application

Application is **Unification** is based on **Relational Join**

A predicate definition's head specifies a relational type which is of the relational product (join)

Calling a predicate is unifying thusly:

- **Joining provided arguments**
- **Matching with column types of predicate definition**
- **Propagating "row" information through this connection in newly expanded tree**

(like fact propagation to query in Prolog)

Measures

Finite countable sets are measured by their counts.

Compact sets are measured by the volume they span,

e.g. intervals by difference of endpoints.

Unbounded countable products of finite countable sets measured by exponentially-decaying weighting on finite set measures,

e.g. string (or sequence) measure is

$\Sigma_{\text{position}}\ 1/|\text{alphabet}|\times 2^{-\text{position}}$

Remaining challenges

Need efficient models of high-dimensional joint spaces (relations)

Strategy so far:

- Find related subspace with *info-clustering*
- Manifold learning within that subspace
- Sketch and train supervised information-gain predictor to give to contextual bandit

Would like these to compose, so considering HRR for manifolds

Tensions

Work on relational algebra is from database world

Databases tend to be aggressively closed-world

We want to be open-world to address AI-like problems

Attempt to unify disparate languages of machine learning, databases, logic programming.