FPLP: A Meta-Interpreter for Fuzzy (Probabilistic) Logic Programming

November 25, 2018

This manual describes how to use FPLP, a meta-interpreter for performing quantitative and/or multi-valued reasoning using tabled logic programming. In FPLP, a ground atom A can be associated with a *strength* σ using the syntax $A \sim \sigma$. What the strength strength represents depends on the actual semantics used. For instance, the stength can be a number that denotes, say, a probability or a value in fuzzy logic. Alternatively, the strength may be an atom denoting, say, a qualitative likelihood such as *"possible"* or *"highly likely"*. Depending on the semantics, the propagation of strengh values through logical expressions and rules varies. The major groups of semantics that are supported are as follows.

- *Lattice-based Semantics* In a semantics bassed on a complete lattice L, each strength s_i is associated with an element of L. The strength of a conjunction $A \sim s_1, B \sim s_2$ is the meet in L of s_1 and s_2, while the strength of a disjunction $A \sim s_1; B \sim s_2$ is the join in L of s_1 and s_2. And finally, the default negation of a literal $naf A \sim s_1$ is complement in A of s_1.

- *T-norm based Semantics.* In a semantics based on a T-norm T (and its associated -conorm) each strength s_i is associated with an element of the real number interval $[0, 1]$. The strength of a conjunction $A \sim s_1, B \sim s_2$ is $T(s_1, s_2)$; the strength of a disjunction $A \sim s_1; B \sim s_2$ is *co-*$T(s_1, s_2)$. And finally, the default negation of a literal $naf A \sim s_1$ is $1 - s_1$.

  As will be discussed below, T-norm based semantics are well-suited for reasoning with vague concepts (such as whether a given person is tall); and with reasoning over knowledge that is based on similarity or relevancy measures. Reasoning about such statements often requires the use of a quantitative strength that is not probabilistic in the formal sense. The current implementation of FPLP supports the min, product, and Lukasiewicz T-norms.

- *Probability-based Semantics.* Last, but not least FPLP also supports probabilistic semantics for logic programs based on the distribution semantics []. Both of these semantics are supported using the syntax of Logic Programs with Annotated Disjunctions (LPADs). The probabilistic semantics supported include:

  - *The full distribution semantics []*, which extends logic programs with full Bayesian reasoning. (In fact logic programs under the full distribution semantics are strictly more powerful than Bayesian nets.) However, like probabilistic reasoning in general, probabilistic logic programming can have a high computational cost.
  - *The restricted distribution semantics.* For applications that don't need the generality of the full distribution semantics, FPLP also offers an im-

plementation of the restricted distribution semantics, which makes the assumption of the independence of occurrence of probabilsitic atoms within a derivation, along with an assumption of the exclusivity of multiple derivations of a subgoal with respect to probabilistic atoms encountered by each derivation.

FPLP implements these extensions within a very general logic programming framework for knowledge representation that includes:

- Specification of fully recursive rules that include both default negation `naf` and explicit negation `neg`.
- The ability to specify the strength of quantitative rules in a variety of functions of the strength of their bodies, including strength boosting, strength decay, and sigmoid functions.
- Evaluation of these rules according to the three-valued Paraconsistent Well-Founded Semantics (WFS)