Specifying Credal Sets With Probabilistic Answer Set Programming

Denis Mauá and Fabio Cozman

Universidade de São Paulo, Brazil

The First Author...

- Denis D. Mauá
  Associate Professor
  Dept. of Computer Science, Institute of Mathematics and Statistics
  ddm@ime.usp.br

Center for AI (C4AI)

- Main funding by FAPESP and IBM.
- About 90 faculty associated, 100 students.

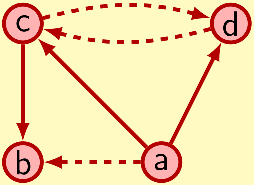
Probabilistic Answer Set Programming

- Logic Programming + Constraint Satisfaction
  + Uncertain Reasoning
  = Probabilistic Answer Set Programming

- Example application: Argumentation under uncertainty (e.g. driving public discourse on climate change).

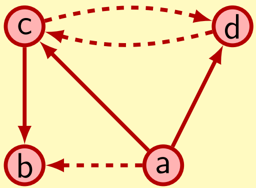
Probabilistic Answer Set Programming

```
0.3::a.
c :- not d.        d :- not c.
c :- a.            d :- a.
b :- a.            b :- not a, c.
```



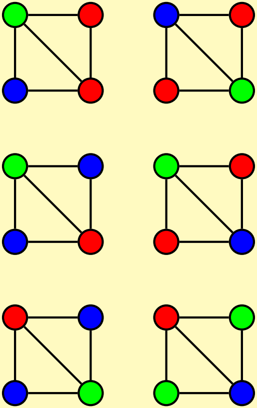
Probabilistic Answer Set Programming

```
0.3::a.
c :- not d.        d :- not c.
c :- a.            d :- a.
b :- a.            b :- not a, c.
```



- Nondisjunctive acyclic ⇒ Bayesian networks.
- Nonstratified or disjunctive ⇒ belief functions.
- This work: How to extend to more general imprecise probability models?

Example: Coloring a random graph



```
node(1). node(2). node(3). node(4).
0.2::edge(1,2). 0.3::edge(2,3).
0.4::edge(3,4). 0.5::edge(1,4).
0.6::edge(1,3).

edge(X,Y) :- edge(Y,X).
conflict(X,Y) :- not conflict(X,Y),
    edge(X,Y), color(X,C), color(Y,C).
color(X,red); color(X,blue); color(X,green) :- node(X).
```

This Work: Extended PASP

Interval-valued PASP

```
[0.1,0.3]::red(X); [0.2,0.4]:: green(X); [0.4,0.6]::blue(X) :- node(X).
```

Parametrized PASP

```
W::win(X); D::draw(X); L::loose(X) :- match(X),
                                      W > D, W > L, L <= 0.3.
```

Expressivity I

Theorem (Standard PASP capture belief functions)

Every **infinitely monotone lower probability over a finite domain** *can be specified by a probabilistic answer set program with precise probabilities in size proportional to the number of focal sets of its m-function characterization.*

Theorem (Interval-Valued PASP capture finite credal sets)

Every **finitely-generated credal set over a finite domain** *can be represented by an acyclic and positive probabilistic logic program with a single vacuous interval-valued annotated disjunction and a set of precise annotated disjunction.*

Expressivity II

Theorem (Simplifying Interval-Valued PASP)

Any interval-valued probabilistic answer set program with interval-valued annotated disjunctions can be converted into an equivalent program containing only interval-valued probabilistic facts (Bernoulli vars) and non-probabilistic rules. If the original program is acyclic (resp., nondisjunctive), the resulting program is also acyclic (resp., nondisjunctive).

Theorem (PASP semantics through credal networks)

The semantics of an **acyclic parametrized probabilistic answer set program** *is given by a credal network; if only probabilistic facts and nonprobabilistic rules appear, the network structure is the dependency graph of the program.*

Complexity

Theorem

*Deciding whether $\underline{Pr}(atom) \geq \gamma$ is NP^{PP}-**complete** in both interval-valued and parametrized probabilistic answer set programs.*

Inference

Compute $\underline{Pr}(a|b)$ by GBR (solve for μ using binary search):

$$\min_{Pr} Pr(a,b) - \mu Pr(b) = 0 \Leftrightarrow \min_{Pr} Pr(a,b) + \mu Pr(\neg b) = \mu$$

augment program with

```
query :- a,b.
mu::query :- not a, b.
```

To conclude...

- Probabilistic Answer Set Programming captures belief functions.
- This work: Extended language to capture any finite credal set.
- Interval-valued PASP implemented in dPASP.
- Challenge: Probabilistic inference is *too* costly.
  - Needed: approximate inference algorithms!
- In the works: connection with neural networks.