Specifying Credal Sets With Probabilistic Answer Set Programming

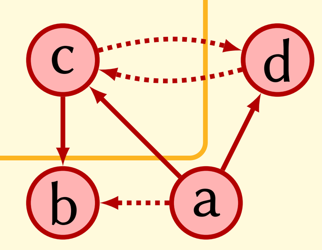
Denis D. Mauá
Universidade de São Paulo, Brazil
denis.maua@usp.br

Fabio G. Cozman
Universidade de São Paulo, Brazil
fgcozman@usp.br

Motivation

- Probabilistic answer set programming provides a flexible and powerful relational language for representing and computing with uncertain knowledge containing recursive definitions, logical constraints and incomparability
  - Sato's semantics describe Bayesian networks and cyclic dependences
  - Credal semantics: describe belief functions by multivalued-mappings
- This work: Extend semantics to general imprecise probability models

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



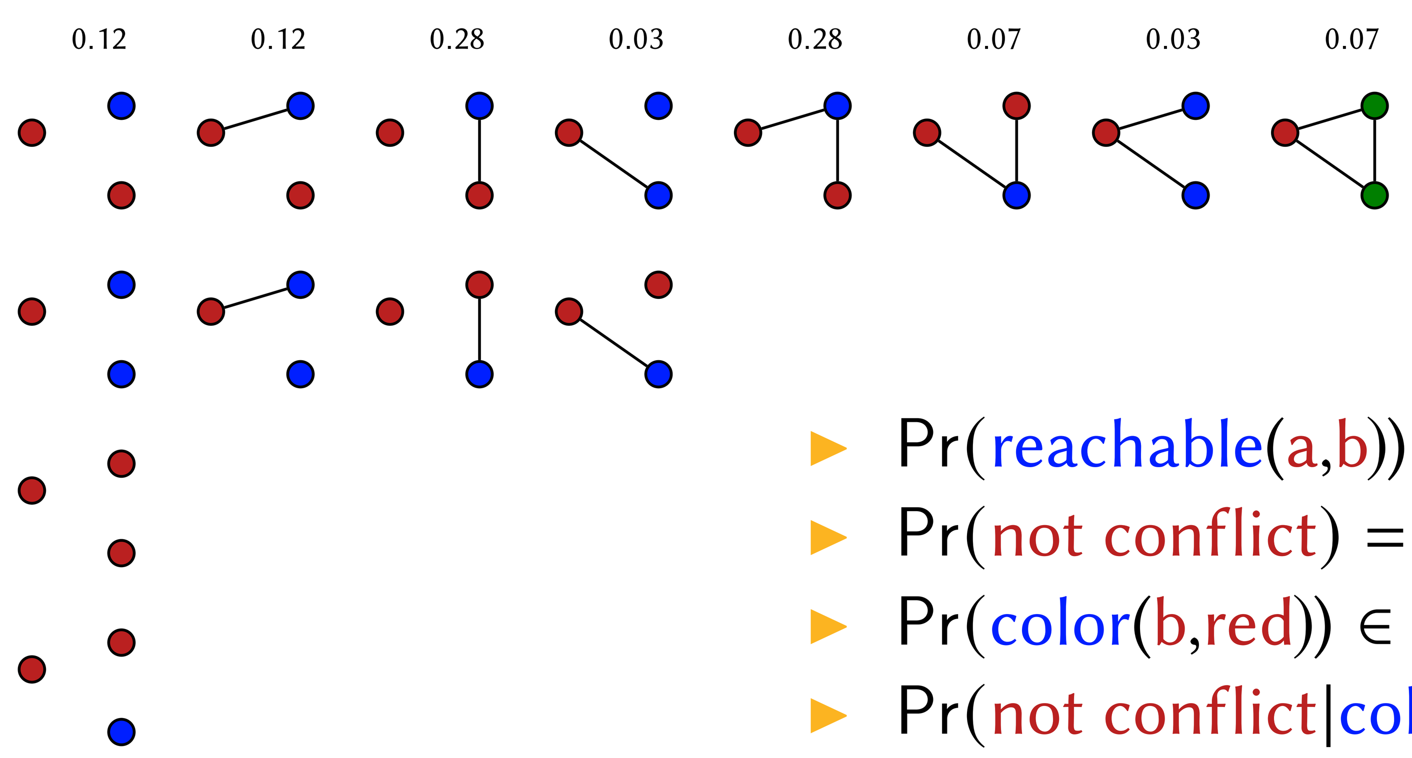
Probabilistic Answer Set Programming

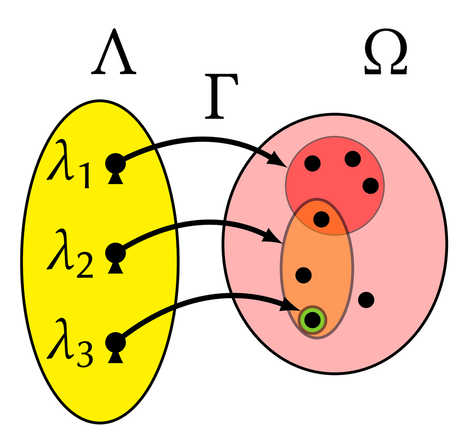
- Atoms represent basic unit of knowledge, e.g. edge(a,b)
- Normal rules constrain models, e.g. edge(X,Y) :- edge(Y,X)
- Disjunctions state incomparable choices, e.g. red(X); blue(X)
- Default negation: not a is true if a cannot be proved
- Annotated disjunctions state probabilistic choices, e.g. 0.2::red; 0.5::green; 0.3::blue
- Stable model semantics: models of the program are minimal models of reduct, where default negation is interpreted away
- Credal semantics: Infinitely-monotone probability induced by:
  - Total choices Λ: selections of independent probabilistic choices
  - Probability mass function $\Pr(\lambda)$ is the product of selected choices
  - Γ maps total choices to stable models of corresponding program

$$\underline{\Pr}(A) = \Pr(\{\lambda \in \Lambda \mid \Gamma(\lambda) \subseteq A\}) \qquad \underline{\Pr}(A \mid B) = \frac{\underline{\Pr}(A \cap B)}{\underline{\Pr}(A \cap B) + \overline{\Pr}(A^c \cap B)}$$

Example

% A random undirected random graph
0.5::edge(a,b). 0.2::edge(a,c). 0.7::edge(b,c).
edge(X,Y) :- edge(Y,X).   color(a,red).
% (recursive) definition of node reachability
reachable(X,Y) :- edge(X,Y).
reachable(X,Y) :- edge(X,Z), reachable(Z,Y).
% (disjunctive) definition of 2–colorability
color(X,red); color(X,blue).
conflict :- edge(X,Y), color(X,C), color(Y,C).
color(X,red) :- node(X), conflict.
color(X,blue) :- node(X), conflict.





- Pr(reachable(a,b)) = 0.57
- Pr(not conflict) = 0.93
- Pr(color(b,red)) $\in [0.14, 0.57]$
- Pr(not conflict|color(b,red)) $\in \left[0.5, \frac{0.5}{0.5+0.07}\right]$

Results: Precise Prob.

Thm. Every infinitely monotone lower probability can be specified by a PASP program in size proportional to the number of focal sets of its m-function characterization.

Proof. Take focal sets $A_1, \ldots, A_n$ and write:

$m(A_1)$::m(1);. . . ;$m(A_n)$::m(n).
x(o1);. . . ;x(ok) :- m(1).
. . .
x(o1);. . . ;x(ok) :- m(n).

where the constants o1, . . ., ok denote the elements of focal set $A_i = \{\omega_1, \ldots, \omega_k\}$. The stable models correspond to the focal sets.

Results: Interval-Valued Prob.

[0.1,0.3]::red; [0.2,0.4]::green; [0.4,0.6]::blue.

Semantics: Extend $\underline{\Pr}_\Gamma(\gamma)$ to $\underline{\Pr}_\Omega(\omega)$

Thm. Every finitely-generated credal set can be represented by an acyclic and positive program with a single vacuous interval-valued annotated disjunction.

Thm. Any program with interval-valued annotated disjunctions can be converted into an equivalent program containing only interval-valued probabilistic facts; if the original program is acyclic (nondisjunctive), the resulting program is also acyclic (nondisjunctive).

Results: Parametrized Prob.

P::win(X); Q::draw(X); R::loose(X) :-
match(X), P > Q, P > R, R <= 0.3.

Expressivity: The semantics of an acyclic PASP 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.

Inferential complexity: Deciding whether the unconditional lower probability of an atom is above a given threshold is NP^{PP}-complete.

This work was partially supported by CNPq, CAPES, FAPESP and IBM.

Center for Artificial Intelligence IME USP