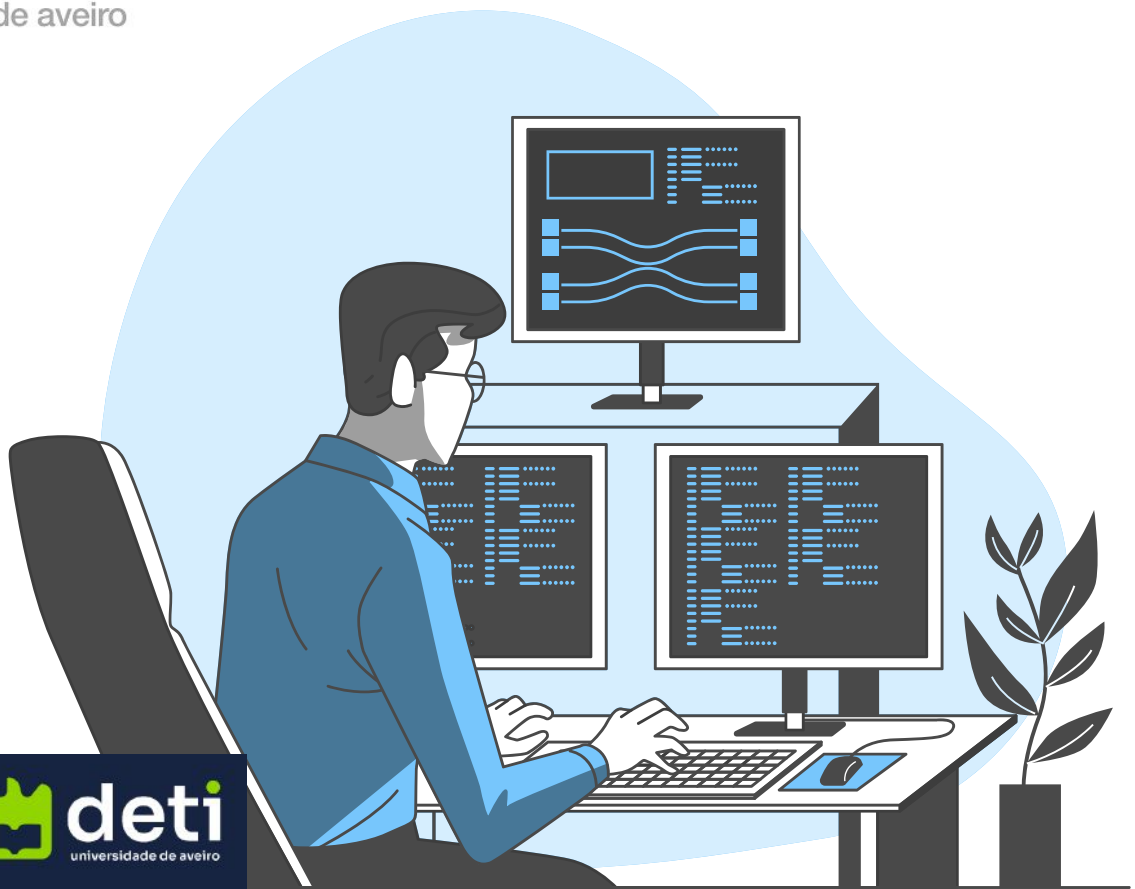
universidade de aveiro



SI-II 24/25

Problog

Study of tools

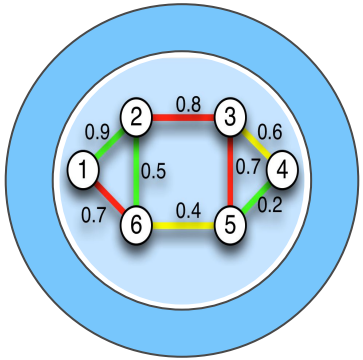
Students:

Artur Almeida - n.º 123196

Pedro Ramos - n.º 107348

deti universidade de aveiro

Introduction



Problog

ProbLog extends Prolog syntax by introducing new operators for probabilistic modeling.

Definition	Example
Fact	raining
Probabilistic fact	0.5 :: raining

Definition	Example
Clause	alarm :- burglary
Probabilistic clause	0.1 :: burglary. 0.9 :: alarm :- burglary.

Key uses

Probabilistic Knowledge Representation

Allows representing uncertain facts and reasoning under uncertainty, making it ideal for domains where data is incomplete or noisy.

AI and Machine Learning Applications

Can be integrated into probabilistic reasoning like Bayesian inferences, Markov Logic, etc.

Robotics and Sensor Networks

Decision-making in uncertain environments, such as estimating the probability of obstacles in robotic navigation based on sensor readings.

Main Features

Probabilistic Facts

Facts have probabilities attached

Probabilistic Inference

Return the probability of a fact being true based on probabilistic rules.

```
1  0.5::rain.
2  0.3::sprinkler.
3
4  0.9::wet_grass :- rain.
5  0.8::wet_grass :- sprinkler.
6
7  query(wet_grass).
```

Rule-Based Probabilistic Reasoning

Reasoning in domains where outcomes depend on multiple factors

This allows the creation of Bayesian Networks and perform real-time calculations on them

Query ▾	Location	Probability
wet_grass	7:7	0.582

Syntax and Semantics

Annotated Disjunction

```
1/6::die(D, 1); 1/6::die(D, 2); 1/6::die(D, 3);
1/6::die(D, 4); 1/6::die(D, 5); 1/6::die(D, 6).
```

- ➔ It expresses that at most one of these choices is true;
- ➔ There is always an implicit null choice which states that none of the options is taken;
- ➔ In this example, however, that extra state has zero probability because the probabilities of the other states sum to one.

...

...

Syntax and Semantics

Queries

```
0.5::heads(C).
two_heads :- heads(c1), heads(c2).
query(two_heads).
```

- A query indicates for which entity we want to compute the probability.
- The resulting value corresponds to the probability that the query succeeds in a randomly sampled program

Query ▾	Location	Probability
two_heads	3:7	0.25

... ...

Syntax and Semantics

Arithmetic Evidence

```
0.5::heads(C).
two_heads :- heads(c1), heads(c2).
evidence(\+ two_heads).
query(heads(c1)).
```

- ➔ Evidence specifies any observations on which we want to condition this probability.
- ➔ Evidence conditions a part of the program to be true or false.

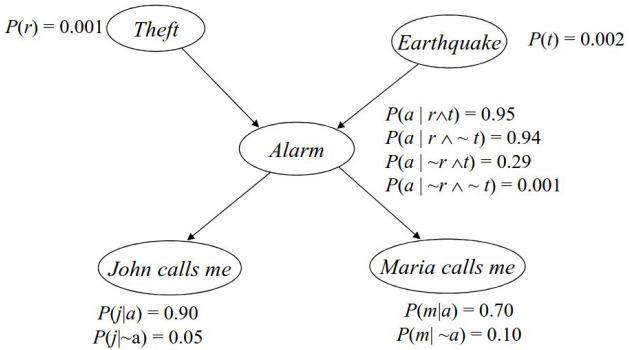
...

Query ▾	Location	Probability
heads(c1)	4:7	0.33333333

.

Example

Bayesian belief networks

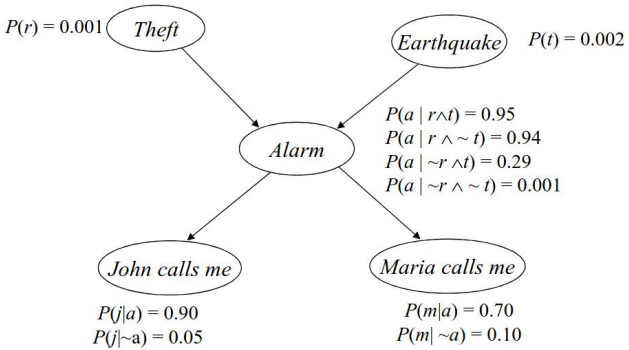



```
0.001::theft.
0.002::earthquake.

0.95::alarm :- theft, earthquake.
0.94::alarm :- theft, \+earthquake.
0.29::alarm :- \+theft, earthquake.
0.001::alarm :- \+theft, \+earthquake.

0.90::john :- alarm.
0.05::john :- \+alarm.

0.70::maria :- alarm.
0.10::maria :- \+alarm.

query(john).
query(maria).
```

Query ▾	Location	Probability
john	15:7	0.052138976
maria	16:7	0.10150987

Example

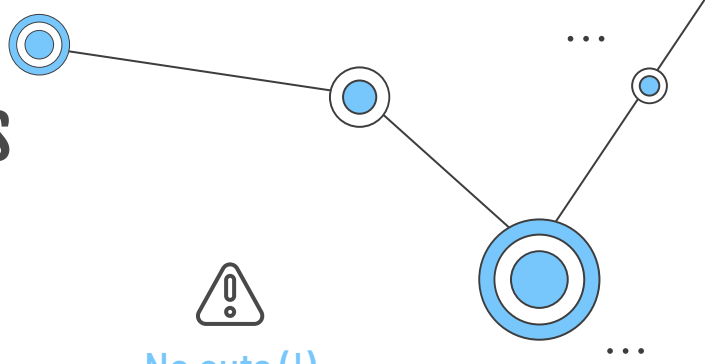
Bayesian belief networks




```
1  0.001::theft.
2  0.002::earthquake.
3
4  0.95::alarm :- theft, earthquake.
5  0.94::alarm :- theft, \+earthquake.
6  0.29::alarm :- \+theft, earthquake.
7  0.001::alarm :- \+theft, \+earthquake.
8
9  0.90::john :- alarm.
10 0.05::john :- \+alarm.
11
12 0.70::maria :- alarm.
13 0.10::maria :- \+alarm.
14
15 evidence(theft).
16
17 query(john).
18 query(maria).
```

Query ▾	Location	Probability
john	17:7	0.849017
maria	18:7	0.664012

Advanced Features



Tabling

Reduce the execution time for exponential growth functions by caching the results of each computation

No cuts (!)

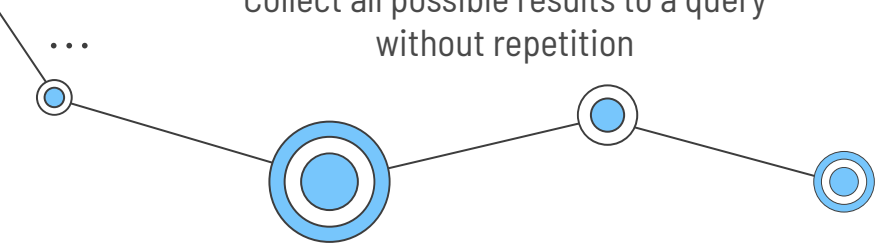
Problog removes the support for features that have no meaning in a probabilistic setting, like cuts (!) or If-Then-Else (->)

Findall

Collect all possible results to a query without repetition

Integration with Other AI Frameworks

With a simple python integration, Problog can be used to solve complex probability networks by other AI frameworks



Example

```
1  :- use_module(library(apply)).
2  :- use_module(library(lists)).
3
4  PH::make_coin(C, PH).
5
6  coin(C) :- make_coin(C, 0.8).
7
8  tonum(C, Num) :- (coin(C), Num=1; \+coin(C), Num=0).
9
10 total(S) :-
11     findall(X, between(1, 10, X), L),
12     maplist(tonum, L, Nums),
13     sum_list(Nums, S).
14
15 query(total(_)).
```

Query ▾	Location	Probability
total(0)	13:7	1.024e-
total(1)	13:7	4.096e-
total(2)	13:7	7.3728e-
total(3)	13:7	0.000786432
total(4)	13:7	0.005505024
total(5)	13:7	0.026424115
total(6)	13:7	0.088080384
total(7)	13:7	0.20132659
total(8)	13:7	0.30198989
total(9)	13:7	0.26843546
total(10)	13:7	0.10737418

How ProbLog Differs from Prolog

Deterministic vs. Probabilistic Logic

Feature	Prolog	Problog
Logic Programming	✅	✅
Deterministic Rules	✅	✅
Probabilistic Reasoning	❌	✅
Handling Uncertainty	❌	✅
Query Results	True/False	Probability Value
AI/ML Applications	Limited	Strong

Demo