Homework 5 (Due April 8)

CS3642: Artificial Intelligence
Kennesaw State University
Spring 2024

In this project, we introduce ourselves to logic programming and probabilistic programming, using ProbLog.

Prolog: Prolog (without the b) is a programming language that is based on first-order logic, which includes propositional logic as a special case. It is based on expressing facts and rules (using logic), and then drawing conclusions from them via a query.

ProbLog: ProbLog is an extension of Prolog that allows one to annotate facts and rules with probabilities. Like Prolog, the language ProbLog is a logic programming language. It is also a probabilistic programming language (see, Church for another example of probabilistic programming language).

The webpage for ProbLog is available at:

`https://dtai.cs.kuleuven.be/problog/index.html`

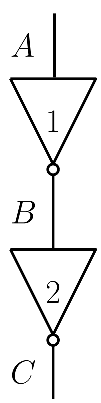
You can install ProbLog using your IDE, or by running the following command on the command line:

```
pip install problog
```

ProbLog also has an online editor where you can edit and run your code online, which is available at:

`https://dtai.cs.kuleuven.be/problog/editor.html`

A simple diagnosis program: Consider the following Boolean circuit:



This circuit consists of two inverters, labeled 1 and 2, and three wires, labeled A, B and C.

Suppose that we model the behavior of this circuit using propositional logic, including the possibility that each inverter is faulty. Consider the following knowledge base for this circuit:

$$\Delta = \begin{cases} \text{ok}_1 \land A & \Rightarrow & \neg B \\ \text{ok}_1 \land \neg A & \Rightarrow & B \\ \neg\text{ok}_1 & \Rightarrow & \neg B \\ \text{ok}_2 \land B & \Rightarrow & \neg C \\ \text{ok}_2 \land \neg B & \Rightarrow & C \\ \neg\text{ok}_2 & \Rightarrow & \neg C \end{cases}$$

The knowledge base Δ is composed of six sentences, three sentences for each inverter. The first two sentences specify the normal operating behavior of an inverter: the output of an inverter is the complement of its input. The third sentence specifies the behavior of an inverter when it is not ok (i.e., faulty): it behaves as if it is stuck at low, i.e., $\neg B$.

Variables ok_1 and ok_2 are called *health variables*, and a term over all health variables is called a *health state*. For example, $\neg\text{ok}_1 \wedge \text{ok}_2$ is a health state where inverter 1 is faulty and inverter 2 is healthy. Given an (abnormal) observation about the system, e.g., $A \wedge \neg C$, we would like to reason about the health states that are consistent with that observation. Any such health state is further called a *diagnosis*.

A simple ProbLog program: The following ProbLog program encodes the behavior of our circuit.

```
0.5::a.
0.9::ok1.
0.9::ok2.

1.0::b :- ok1,\+a.
0.0::b :- ok1,a.
0.0::b :- \+ok1.

1.0::c :- ok2,\+b.
0.0::c :- ok2,b.
0.0::c :- \+ok2.

query(c).
```

The first three lines specifies the three inputs to our problem, which is the value of wire A, and the health of the inverters ok_1 and ok_2. We use three propositional variables `a`, `ok1` and `ok2` for this purpose. If variable `a` is true, that means the value of the wire is high. Otherwise, if it is false, then the value of the wire is low. If variable `ok1` or `ok2` is true, then it means the corresponding inverter is healthy. Otherwise, it is faulty. We have also assumed here that the wire A is high with probability 0.5. Similarly, each inverter is healthy with probability 0.9. These first three lines are called (probabilistic) facts.

The second three lines specifies the behavior of the first inverter, specified as rules of the form `p::x :- y`. You can read this as "if y is true, then x is true, with probability p." The first line says "if inverter 1 is healthy, and the wire A is low, then the wire B is high, with probability 1." Note that `\+a` means the negation of `a`. The second line says "if inverter 1 is healthy, and the wire A is high, then the wire B is high, with probability 0." Equivalently, this just says that "if inverter 1 is healthy, and the wire A is high, then the wire B is low." (Prolog syntax does not allow us to directly say that wire B is low, so instead we specify a rule with zero probability). The third line says "if inverter 1 is faulty, then the wire B is high, with probability 0," i.e., the wire B must be low in this case.

The next three lines specifies the behavior of the second inverter.

The last line `query(c)` specifies that we query the probability of variable c, i.e., the probability that the output wire C is high. If you copy-and-paste the above code in the online ProbLog editor, and click the "Evaluate" button, you should find that the probability of `c` is 0.495. Assuming that both inverters are healthy, the probability that the output wire is high should be 0.5, if the probability that the input wire is high is also 0.5. However, given the possibility of a fault, the probability that the output wire is high is slightly less than 0.5.

Answer the following questions about our simple circuit diagnosis problem.

1. Replace our query above (the last line), with the following lines:

```
evidence(a,true).
evidence(c,true).
query(ok1).
query(ok2).
```

Which asserts that wire A is high and wire C is high, which corresponds to normal operating behavior. We also query the probabilities that inverters 1 and 2 are healthy, which should be 0.9 (for inverter 1) and 1.0 (for inverter 2). If we have a 1.0 probability that inverter 2 is healthy, then that means that it is impossible that inverter 2 to be faulty. Why is it impossible for inverter 2 to be faulty, in terms of the (logical) model of our circuit?

2. Replace our query now with the following lines:

```
evidence(a,true).
evidence(c,false).
query(ok1).
query(ok2).
```

In this case, if wire A is high and wire C is low, then this corresponds to a faulty operating behavior. What is the probability that inverter 1 is healthy, and the probability that inverter 2 is healthy?

3. Replace our query now with the following lines:

```
evidence(a,false).
evidence(c,true).
query(ok1).
query(ok2).
```

In this case, if wire A is low and wire C is high, then this corresponds to a faulty operating behavior. What is the probability that inverter 1 is healthy, and the probability that inverter 2 is healthy?

Turn in: the observations requested above as a text file or as a .pdf file (no Microsoft Word .doc's please), onto the course website under `Assignments` and `Homework 5`. Assignments are due Monday, April 8 by 11:59pm. Please start early in case you encounter any unexpected difficulties. Late assignments are accepted without penalty until solutions are posted.