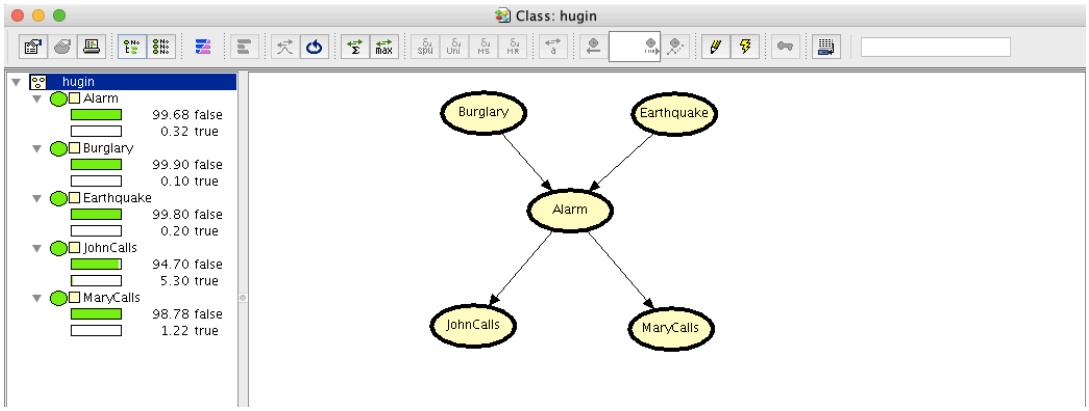
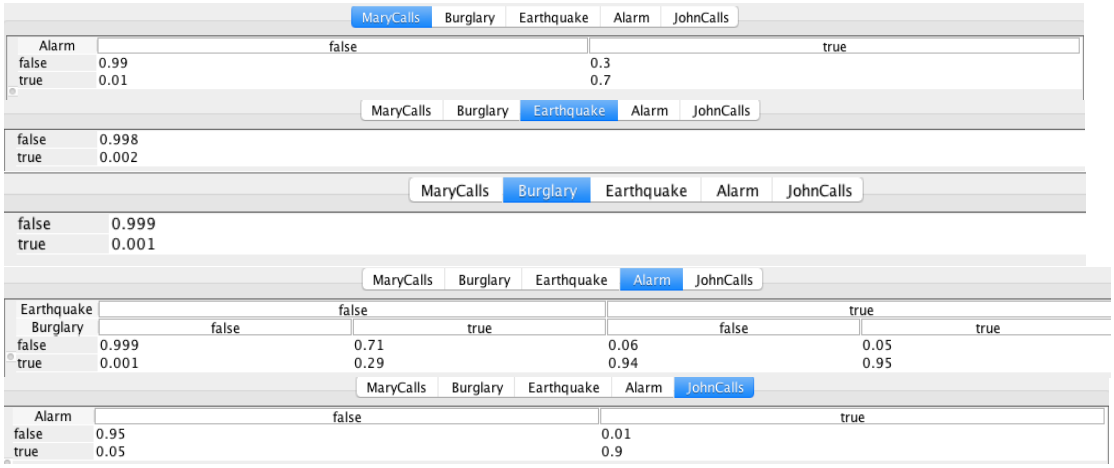
Laura Daniela Vargas Herrera A01203961
María Fernanda Vargas Herrera A01203957
Bayes Network



We chose the Alarm example for comparing our solution with Hugin tool. The software gives us the probability table for each node. Those are shown below:



Some examples:

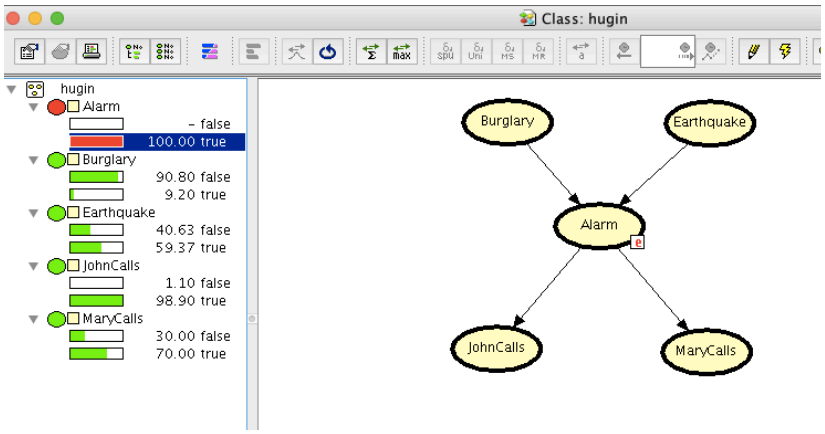
What is the probability of Mary calling given that the alarm rang? (+MaryCalls | + Alarm)
The result obtained with Hugin is 70% which is correct.

Code:

```
+MaryCalls|+Alarm
0.7
```

Hugin:

Laura Daniela Vargas Herrera A01203961
María Fernanda Vargas Herrera A01203957
Bayes Network

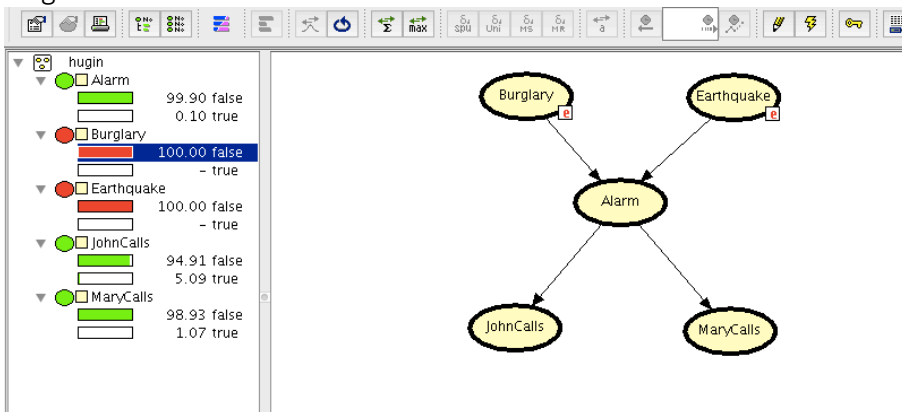


Another example, what is the probability of the alarm rand given that there was not earthquake and neither Burglary? (+Alarm | - Earthquake, - Burglary) the result is .10%

Code:

```
+Alarm|-Earthquake,-Burglary
0.001
```

Hugin:

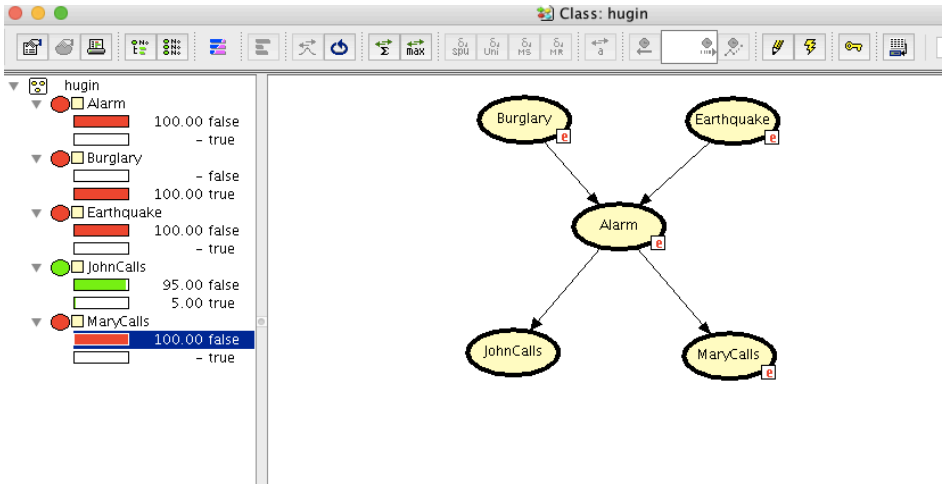


We also calculated with Hugin, what is the probability of John calling if there was no alarm, no earthquake, Mary didn't call but there was a Burglary? (-JohnCalls | -Alarm, + Burglary, -Earthquake, -MaryCalls) the values is 95% which coincides with the output of our program.
Code:

```
-JohnCalls|-Alarm,-Earthquake,-MaryCalls,+Burglary
0.95
```

Hugin:

Laura Daniela Vargas Herrera A01203961
María Fernanda Vargas Herrera A01203957
Bayes Network



Hugin is a software that lets you obtain the probabilities in a friendly way making it visual and easier to understand. With the software creating the Bayes Network was simple as all you needed to do was creating the nodes and assigning the probabilities and the relations between the other nodes. To calculate the probability, you only must select the values (true or false) for each evidence. Once you select the evidence the probabilities of the nodes that have a relation with the selected node will be affected and you can see this in the interface. As you can see it in the examples.

Our solution is a simple python program that calculates the probabilities given a Bayes Network. The input is divided in three parts: Nodes (all the nodes in the Bayes Network), Probabilities for each node given its parents, and the last input are the queries to calculate. The result is the probability for each number as a decimal.

The main difference between our solution and Hugin, in Hugin you can see all the probabilities that are affected depend on the query, while in out solution we are just giving the result of the query. Another difference is that Hugin's solution is given in percentage while ours in decimal (90%, .90). Hugin shows the Bayes Network graphically making it easier to understand the relations between the nodes, and to determine which nodes are being affected by the given one.

Both are based on Bayes Network; the implementation might be different. Our algorithm visits all the parents of the node to get the result while Hugin must have a better algorithm that simplifies and optimizes the calculations. They can be used to get the probability of values that cannot be observed.

In real life application like medical diagnosis, and stock market in which both cases must be really precise, and the probability should be frequently calculated to make the best decisions, those cases will need a professional, advanced and automated tool like Hugin. Hugin experts has three different solutions such as Bayes Fraud used during claims handling for fraud detections, Bayes AML used for banks, casinos and other risk entities, and the last one, Bayes Credit that identifies and manages credit default risks. The Hugin we used if the lite version so you cannot create a Bayes Network with a huge number of nodes, but it exits a paid version where you have unlimited resources to create the Bayes Network.