Hugo Ricardo De la Rosa Coronado - A10208312
Carla Fernanda Estrada Romero - A01203458

Lab 4 Bayesian Networks

When you've finished your program, you can compare your implementation to Hugin Lite. Hugin lite is a software kit for developing graphical Bayes networks. Download it here.

Create your own example network based on a real example. Make it between 3 and 4 nodes. Wikipedia has a good example. You'll have to come up with your own though!

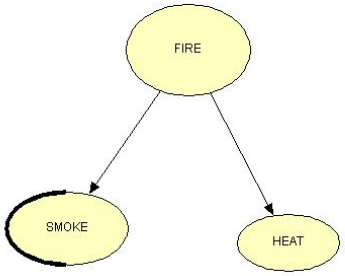
The example that we choosed is about acknowledging fire from smoke and heat, so from heat or smoke we could calculate the probability of a fire in a place or not.

This example will be used to compare the results from the code that we just had build and the Hugin software. The Hugin software is a GUI that helps to relate nodes and describe the probability that are part of a Bayesian Network. They as a company have many solutions based on bayes networks that they have builded for identifying Frauds or even determining if a person is a good candidate for a loan.

The example was encoded using the notation like this:

```
Fire,Smoke,Heat
5
+Fire=0.1
+Smoke|-Fire=0.001
+Smoke|+Fire=0.90
+Heat|+Fire=0.99
+Heat|-Fire=0.0001
3
+Fire,+Smoke
+Smoke
+Fire|+Smoke
```

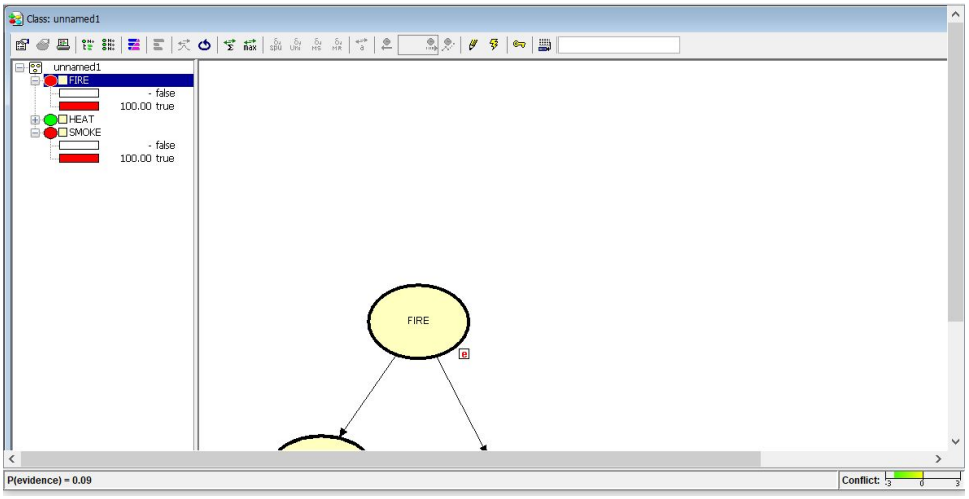
Where the first row, describes the nodes that are going to be used in the network, the next 5 rows will indicate the way that this "playing" nodes are related with their probabilities. And in a graphic way this will look like:



This means that the fire is the node that given, will make that smoke and heat will be considered as conditionally independent.

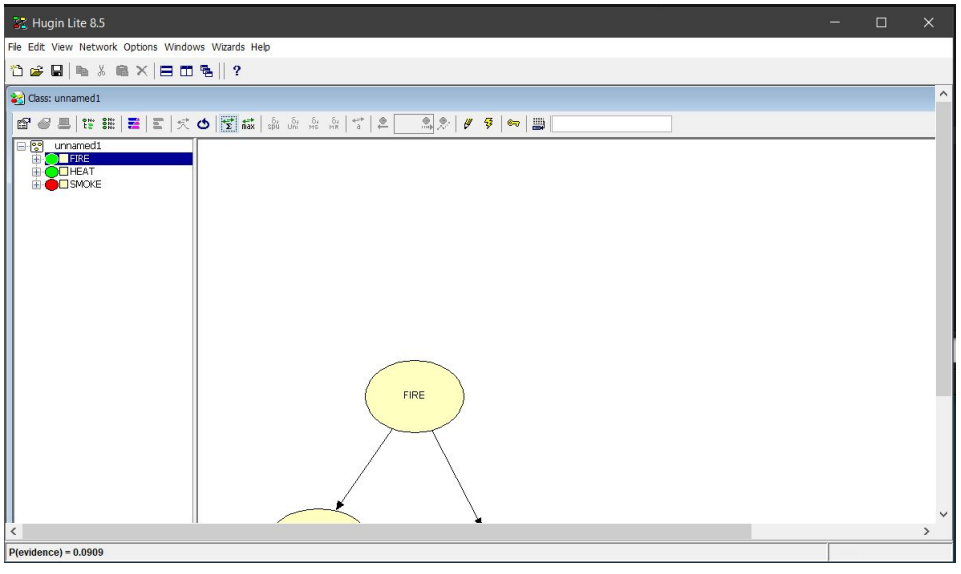
The results are compared using the HUGIN software and the program we just created.

QUERY	HUGIN	OUR'S
+Fire,+Smoke	0.09	0.09



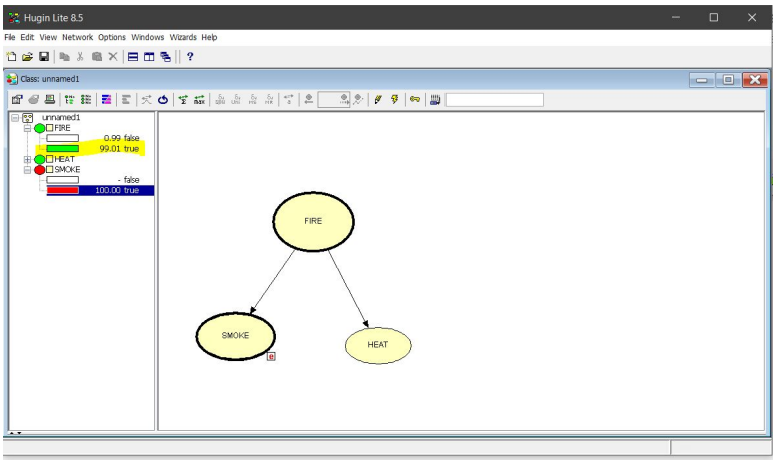

```
muxhingon_1:~/workspace $ ./a.out < in3
0.09
0.0909
0.9901
```

QUERY	HUGIN	OUR'S
+Smoke	0.0909	0.0909




```
muxhingon_1:~/workspace $ ./a.out < in3
0.09
0.0909
0.9901
```

QUERY	HUGIN	OUR'S
+Smoke	0.9901	0.9901



The results from this bayesian network in both programs were the same, they are using the same method to obtain the probabilities for doing the queries. We investigated a little more with other examples and found out that there is a difference. Reading in some forums, people explain than even the computer architecture could influentiate in the result, but not in an extreme way since they have the same statistical basis for approaching the problem.

In this case we didn't found the exact implementation that Hugin is using, but we could infer that they are using the joint probability and the chain rules to obtain the results; because that are the ones that we are using in our program and the results are the same.

In real life applications, we would use the Hugin software, because it has networks already been tested by them and feeded by massive data and a whole strategic team behind the development of such tool. For instance we could try to make our own company like Hugin using the appropriate information and building the appropriate network for selling it.

The usage of this kind of tools in the company environment is really useful for avoiding certain risks that are presented in different situations to the companies and could avoid the loss of several time and money.

Also it's real that we could use our developed program for building new networks and try how they work; and for business is a risk that we have to take into account that the network, or any of its components is not properly setup, which doesn't mean the program is wrong, but we would need probabilistic experts from field to determine the values that are going to be used to be evaluated.