Meeting 9:
Bayesian (decision) networks

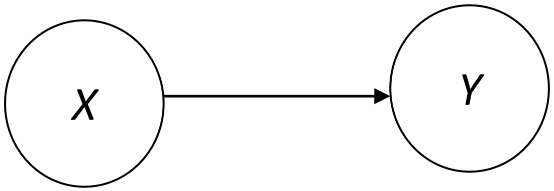
Bayesian (belief) networks

A *Bayesian network* is a connected directed acyclic graph (DAG) in which

• the nodes (vertices) represent *random variables*

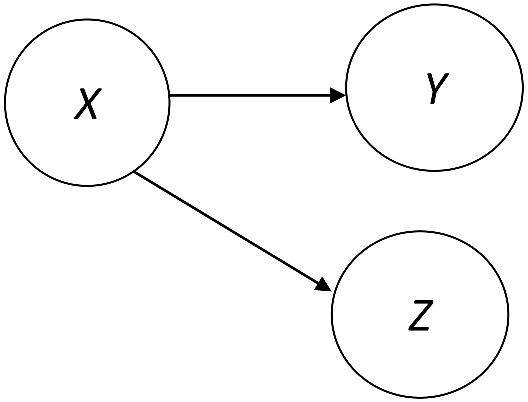
• the links (edges, arcs) represent direct *relevance* relationships among variables

Examples:



This small network has two nodes representing the random variable X and Y.

The directed link gives a relevance relationship between the two variables that means $\Pr(Y = y \mid X = x, I) \neq \Pr(Y = y \mid I)$



This network has three nodes representing the random variables X, Y and Z.

The directed links give relevance relationships that means

$$\Pr(\,Y = y \mid X = x, I\,) \neq \Pr(\,Y = y \mid I\,)$$

$$\Pr(\,Z = z \mid X = x, I\,) \neq \Pr(\,Z = z \mid I\,)$$

but also (as will be seen below)

$$\Pr(\,Z = z \mid Y = y, X = x, I\,) = \Pr(\,Z = z \mid X = x, I\,)$$

Probability "tables"

Each node represents a random variable.

This random variable has *either* assigned probabilities (nominal scale or discrete) or an assigned probability density function (continuous scale) for its states.

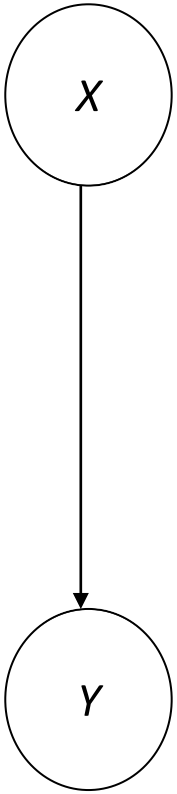
For a node that is *solely* a parent node:

The assigned probabilities or density function are conditional on background information only (may be expressed as unconditional or *prior* probabilities)

For a node that is a child node (solely or joint parent/child):

The assigned probabilities or density function are conditional on the states of its parent nodes (and on background information).

Example:



X has the states x_1 and x_2

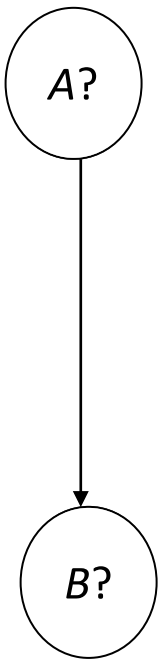
Y has the states y_1 and y_2

Probability tables

X	*Probabilities*
x_1	$\Pr(X = x_1 \mid I)$
x_2	$\Pr(X = x_2 \mid I)$

		Probabilities	
X:		x_1	x_2
Y:	y_1	$\Pr(Y = y_1 \mid X = x_1, I)$	$\Pr(Y = y_1 \mid X = x_2, I)$
	y_2	$\Pr(Y = y_2 \mid X = x_1, I)$	$\Pr(Y = y_2 \mid X = x_2, I)$

Example Dyes on banknotes (from previous lectures)



Two states:

A: "Dye is present"

$\bar{A}$: "Dye is absent"

A?	*Probabilities*
A	0.001
$\bar{A}$	0.999

Two states:

B: "Result is positive"

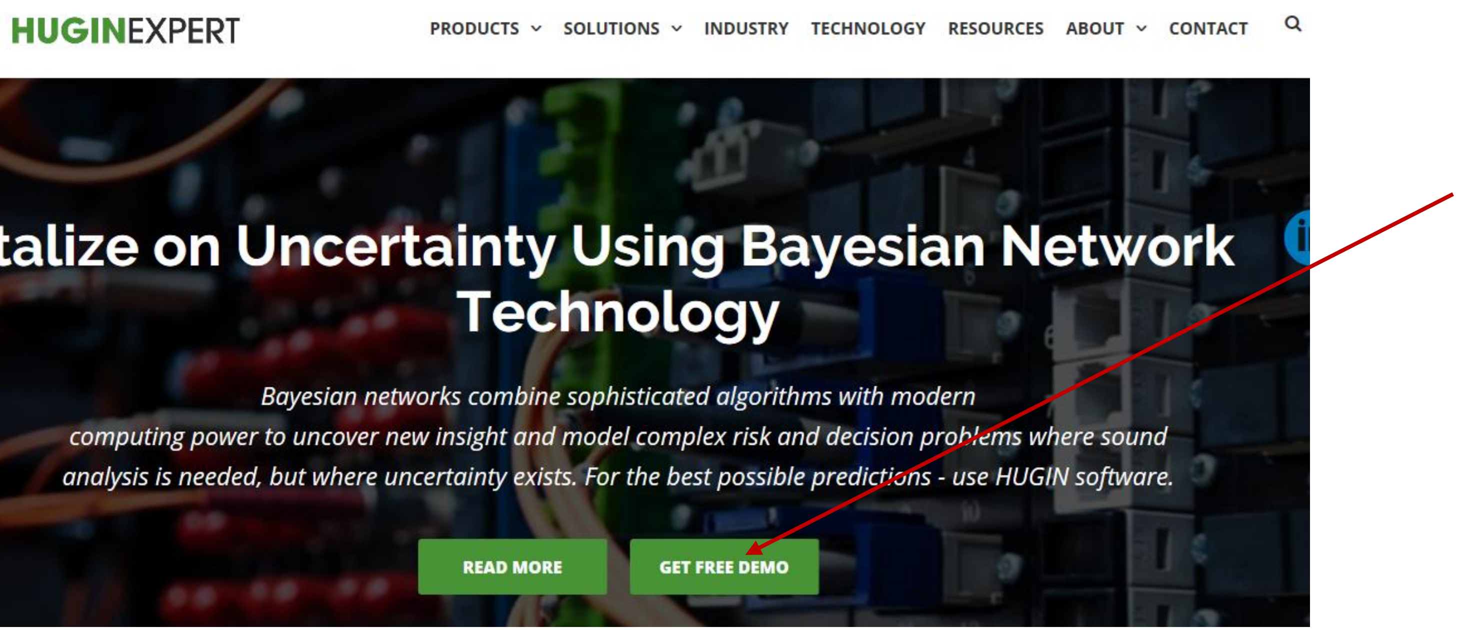
$\bar{B}$: "Result is negative"

		Probabilities	
A?:		A	$\bar{A}$
B?:	B	0.99	0.02
	$\bar{B}$	0.01	0.98

Software

- Hugin (several types of commercial licenses available), Hugin Lite as demo version free of charge
- GeNIe (https://dslpitt.org/dsl/genie_smile.html) used to be easy download freeware, but today it is more complicated
- Agena Risk (https://www.agenarisk.com/) , trial version can be downloaded, otherwise commercial license needed
- *…several other*

Hugin (www.hugin.com)



Download the FREE HUGIN Lite demo

How to get started with **HUGIN** software

The best way to learn more about the HUGIN technology is to try it yourself. Download our free HUGIN Lite demo, a limited version of HUGIN Developer / Researcher.

The HUGIN Decision Engine in HUGIN Lite is available with interfaces for four different programming environments: C, C++, .NET, Java, and as an ActiveX-Server for Visual Basic.

It is prohibited to use the free HUGIN Lite for any other purpose than the demonstration of capabilities and proof of concept.

Go to our Technology Site to learn more about HUGIN Bayesian network technology.

Download Free Hugin Lite – Windows

Download Free Hugin Lite – Linux

Download Free Hugin Lite – Mac OS X

HUGIN app is now available on Google Play and App Store

GET IT ON Google Play

Hent i App Store

Fields marked with an * are required

Name *

Organisation

Address

City

Postal Code

Country

Denmark

Email *

Use of demo

Industrial application

Learn Hugin

Surfing around

DOWNLOAD

Use of demo

Industrial application
Medical application
Financial application
Research
Education
Just for fun
Other

DOWNLOAD

GET IT ON Google Play

Hent i App Store

Dear Anders Nordgaard,

Thank you for downloading the HUGIN Lite Evaluation.
The registration key required for installing this product and the links for download this product have already been sent to your email.
Please check your email to download it.
Thank you.

Thank you for downloading the HUGIN Lite demo.

The registration key required for installing this product is:

- Email: anders.nordgaard@liu.se
- Key: 2384

Information to be used during installation

Download URLs:

32-bit:

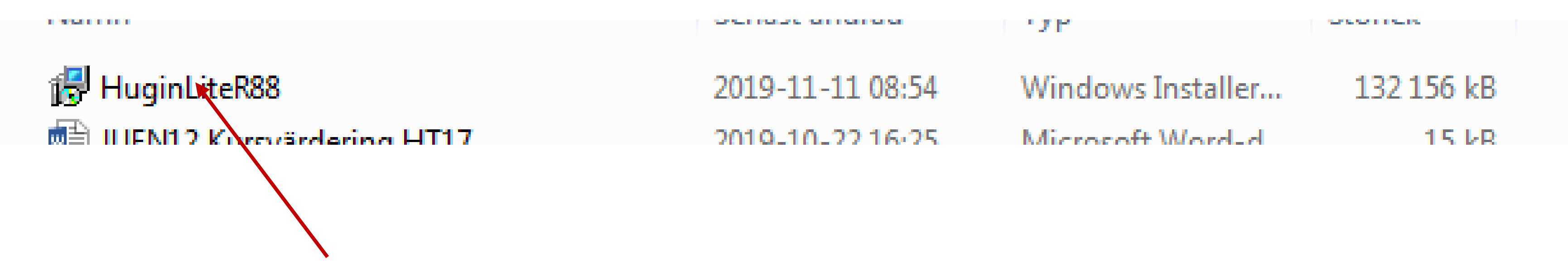
- http://download.hugin.com/pub/Licenses/8.8/HuginLiteR88.msi

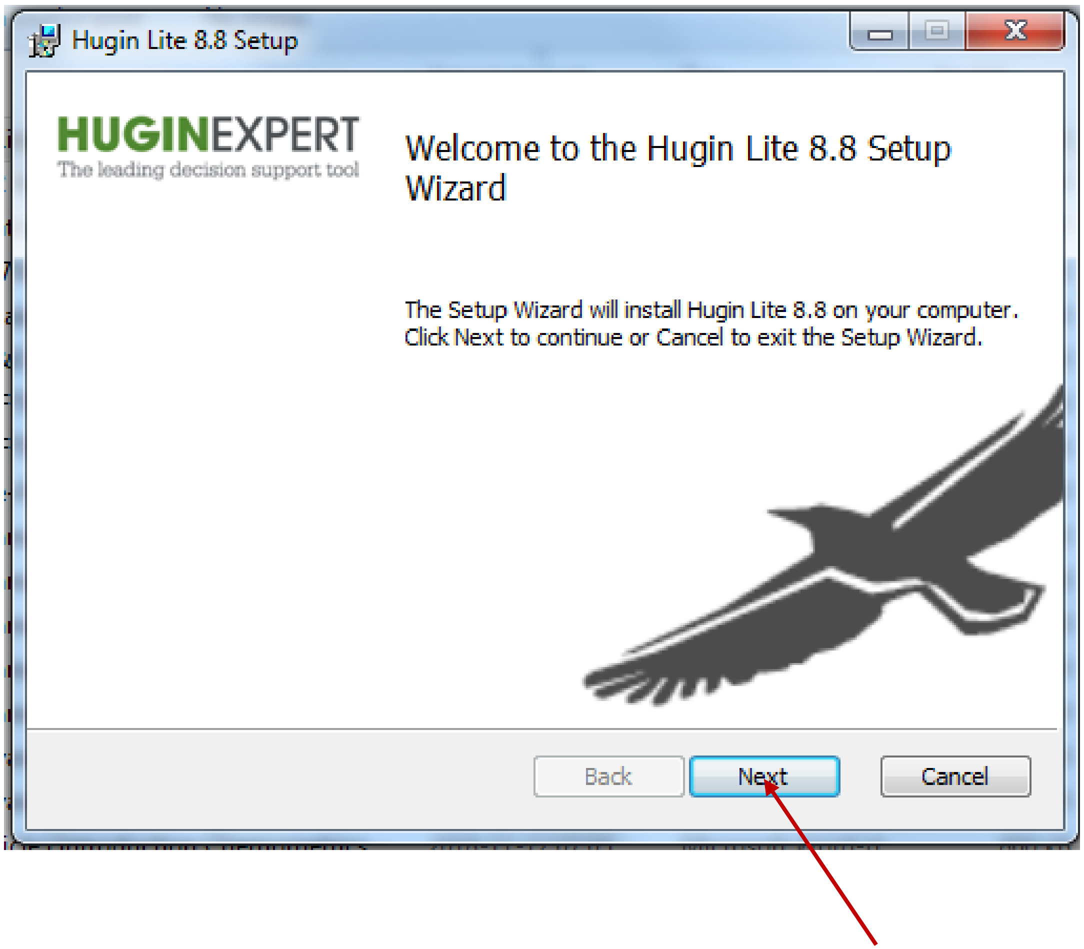
64-bit:

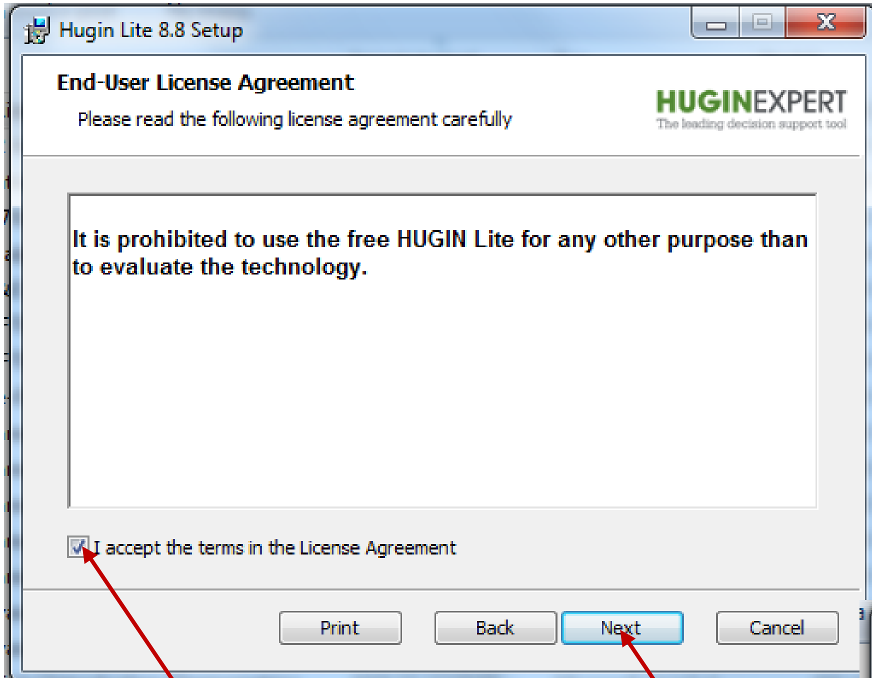
- http://download.hugin.com/pub/Licenses/8.8/HuginLiteR88(x64).msi

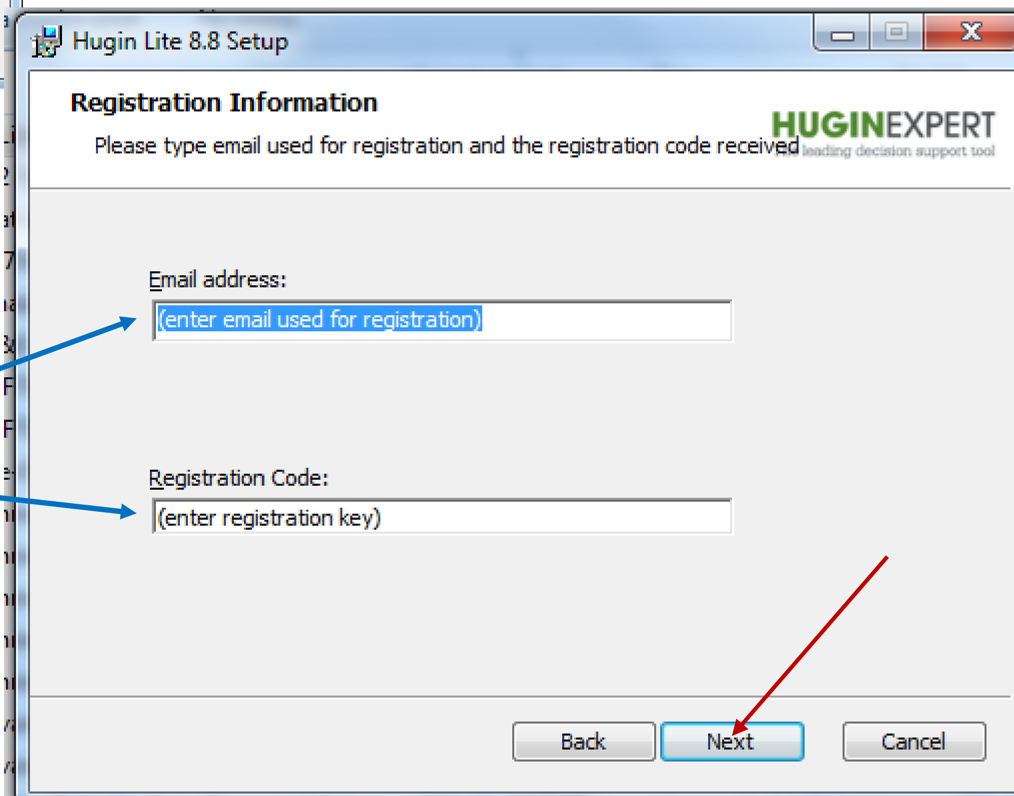
We hope you will find the product useful. If you need a quote for the full license, please contact sales@hugin.com

Best regards,
HUGIN EXPERT A/S
info@hugin.com

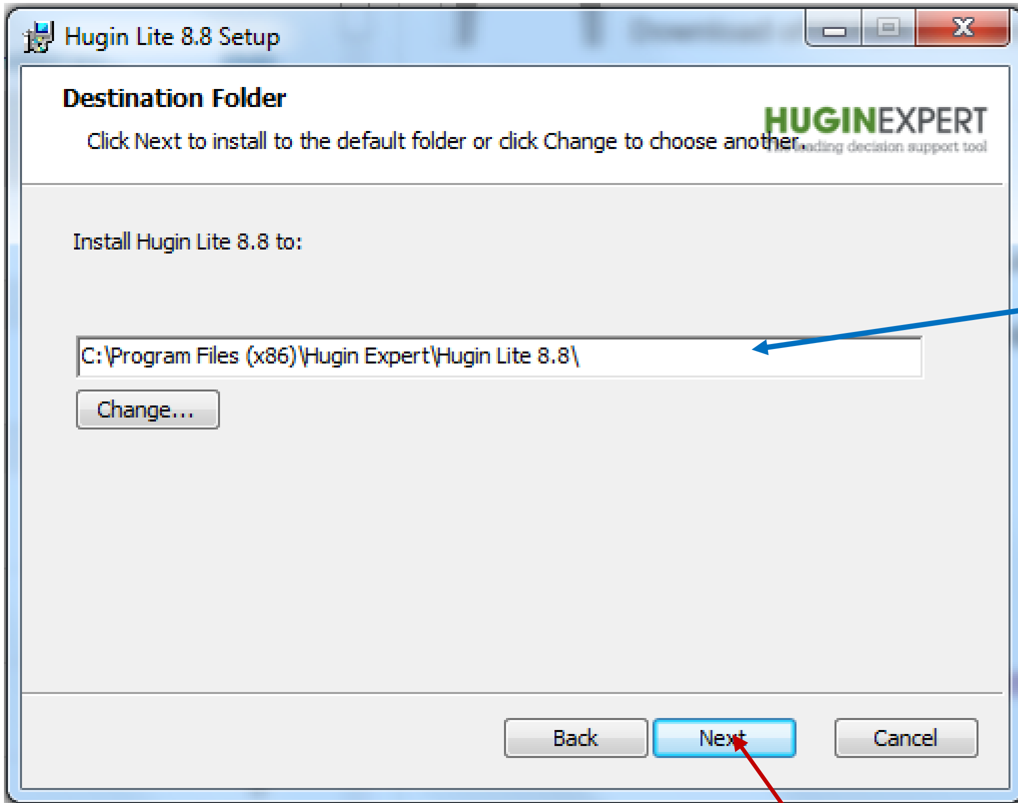

HuginLiteR88
2019-11-11 08:54
Windows Installer...
132 156 kB


Hugin Lite 8.8 Setup
HUGINEXPERT
The leading decision support tool
Welcome to the Hugin Lite 8.8 Setup Wizard
The Setup Wizard will install Hugin Lite 8.8 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel

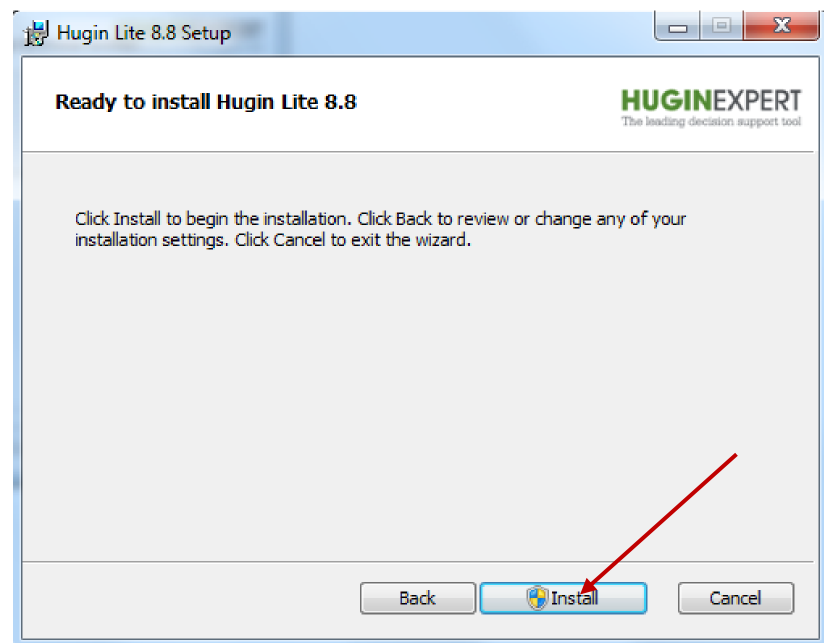




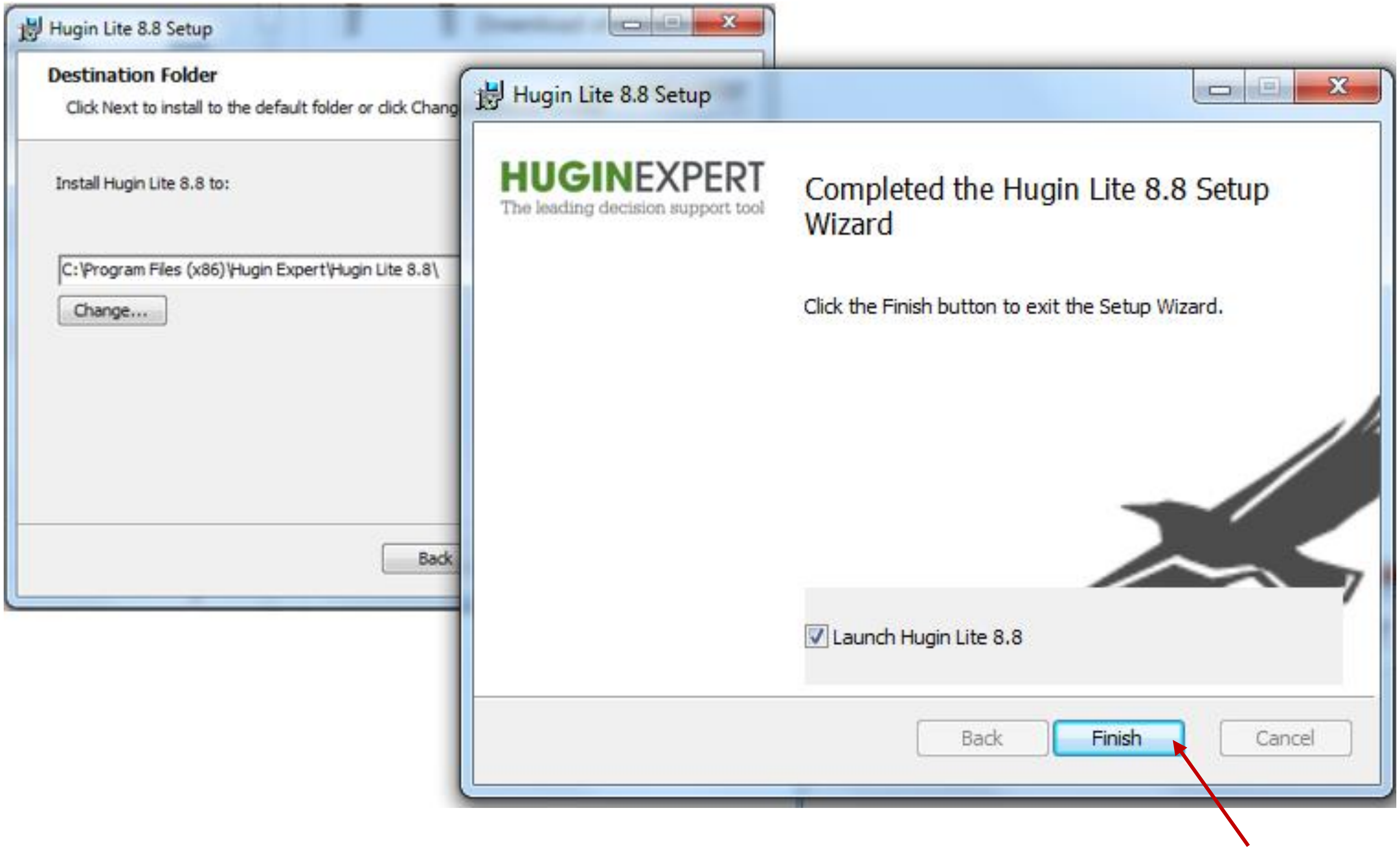
Enter information from your email from Hugin Expert



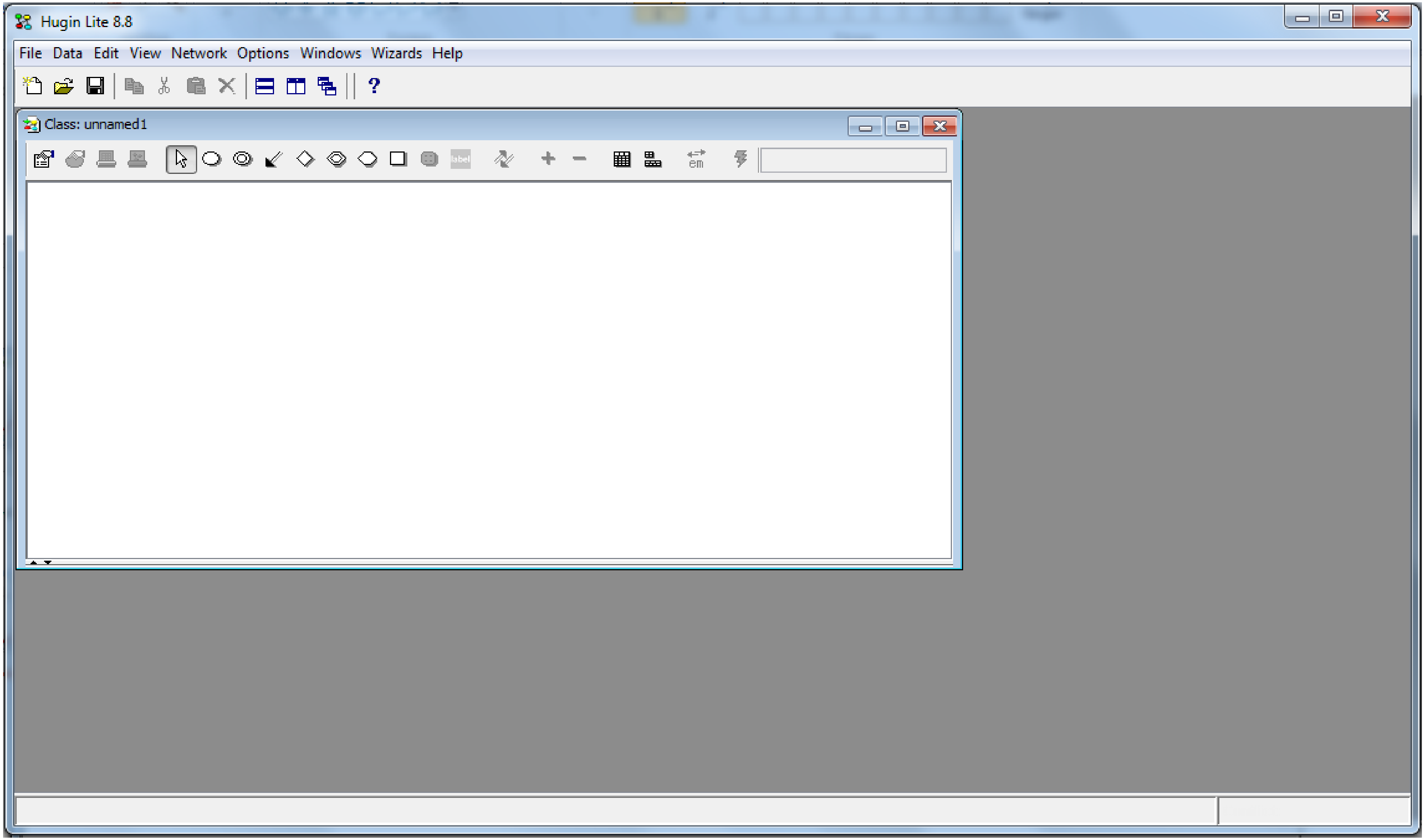
Use this or your own preferred folder



Installation takes less than one minute on most computers.



Upon launching (or opening):



Example to work with

Let A be a random node with states

A_1 = "Willie is a cat"

A_2 = "Willie is a parrot

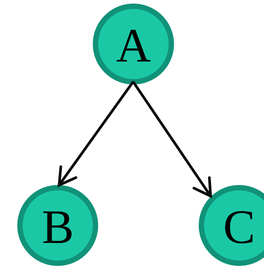
Let B be a random node with states

B_1 = "Willie has four legs"
B_2 = "Willie has two legs"

Let C be a random node with states

C_1 = "Willie has a beak" (*bill*, *nib*)
C_2 = "Willie has no beak"

***Divergent** connection*

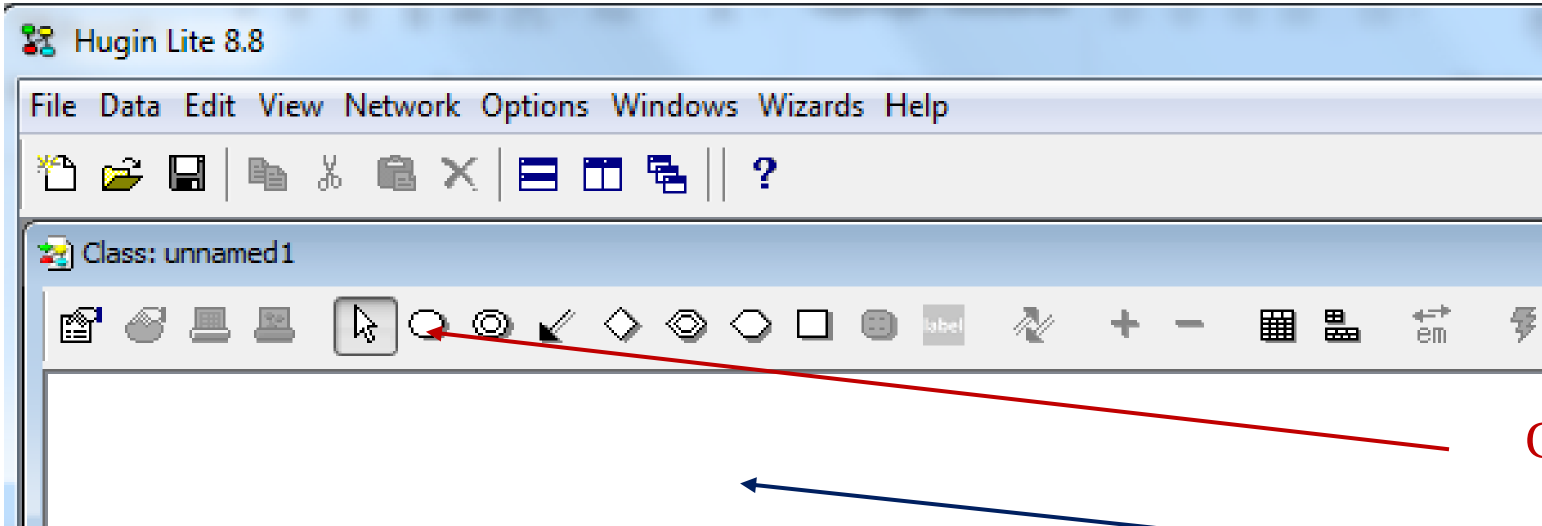


Given B being equal to B_1 the conditional probability of C_1 is different (lower) than the conditional probability of C_1 given B is equal to B_2.

Hence, B is relevant for C and vice versa.

However, if A is given to be A_2, i.e. Willie is a parrot, B and C are no longer relevant for each other if we reasonably assume that the number of legs a parrot has cannot affect whether he has a beak or not.

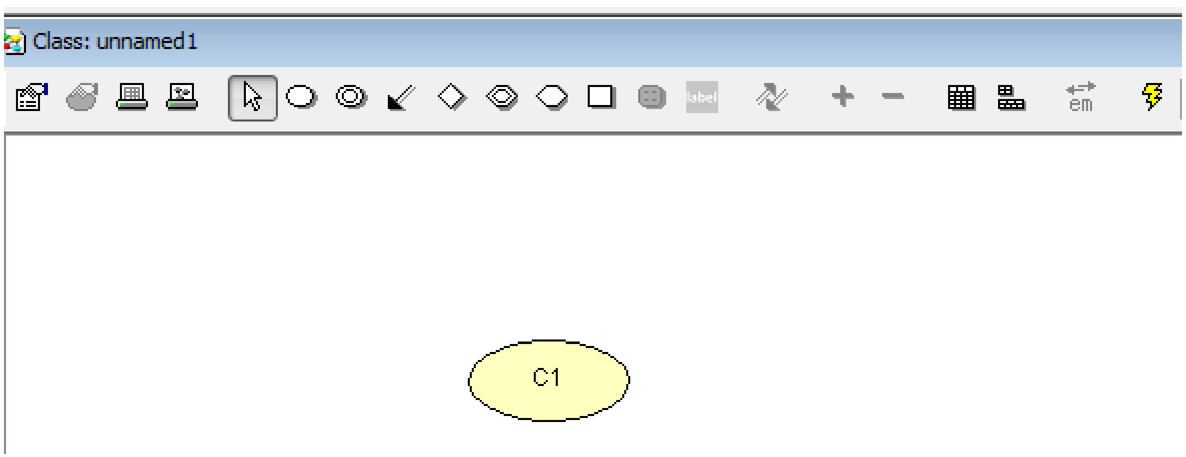
Adding a chance node

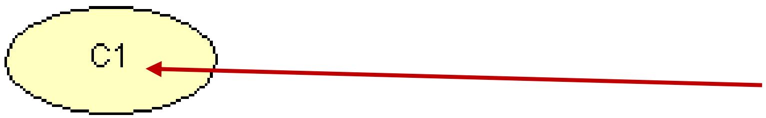


Click here

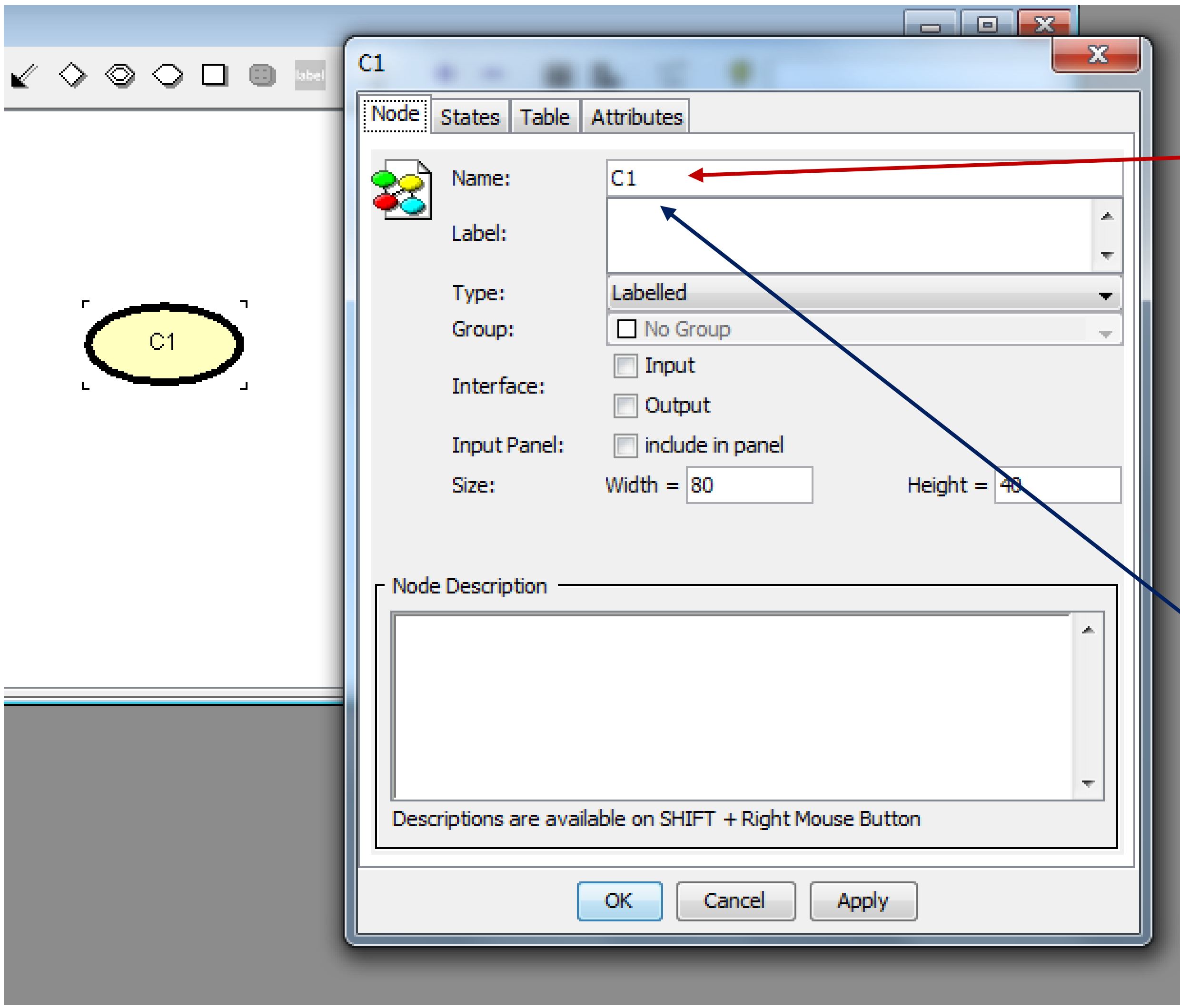
then

Click here





Double-click



Enter a unique name (identifier)

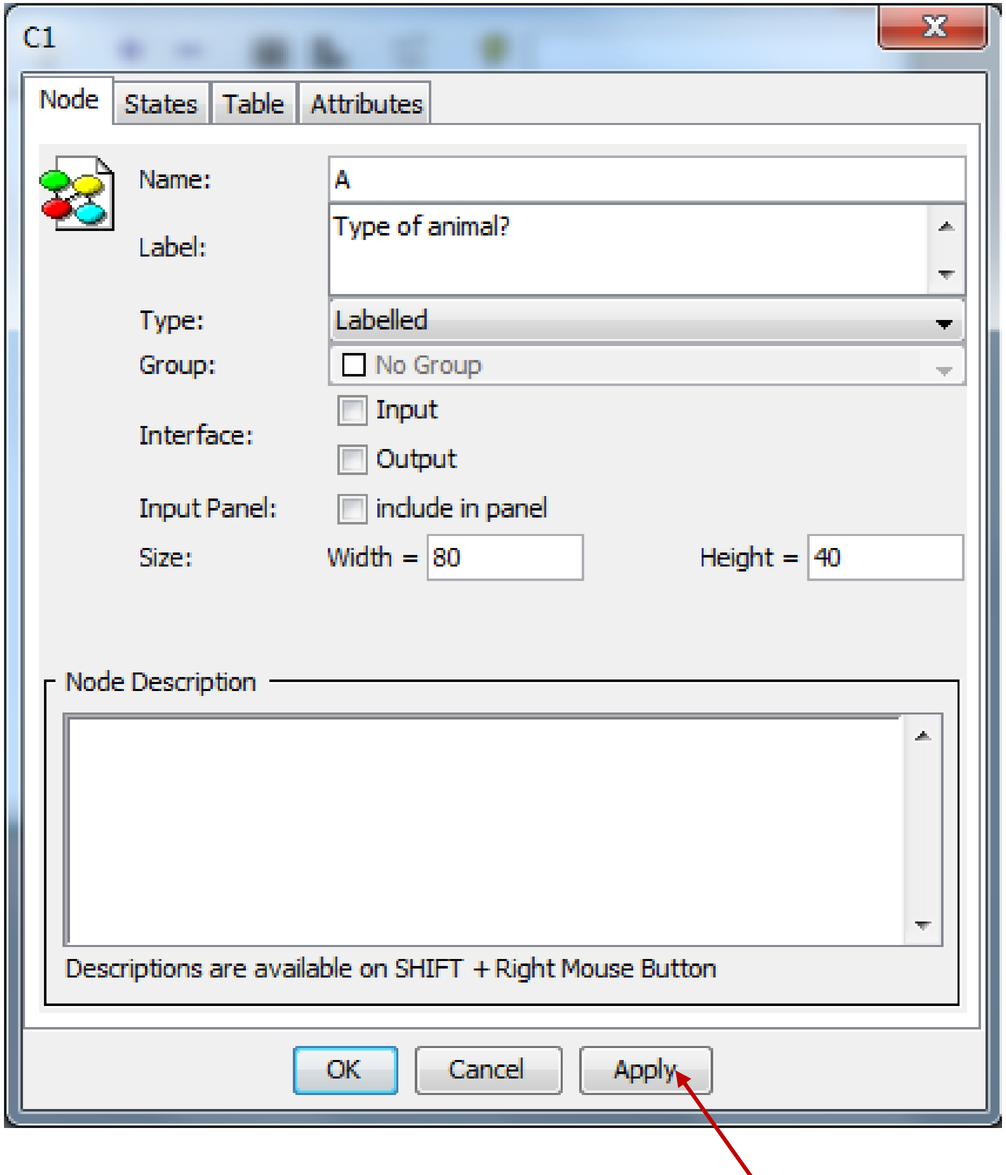
(One single word starting with a letter and otherwise comprised by letters, digits and underscores only)

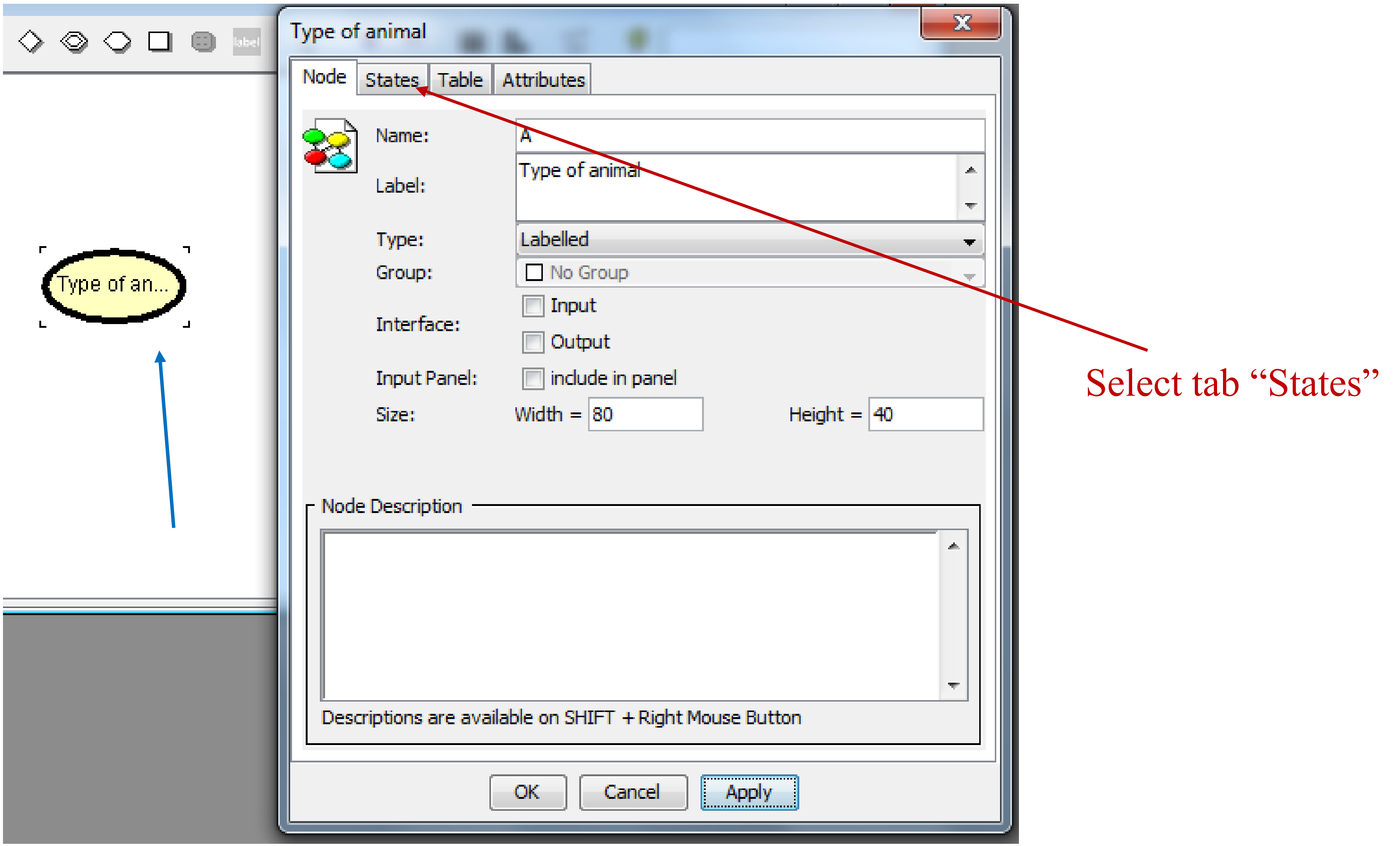
Enter a label

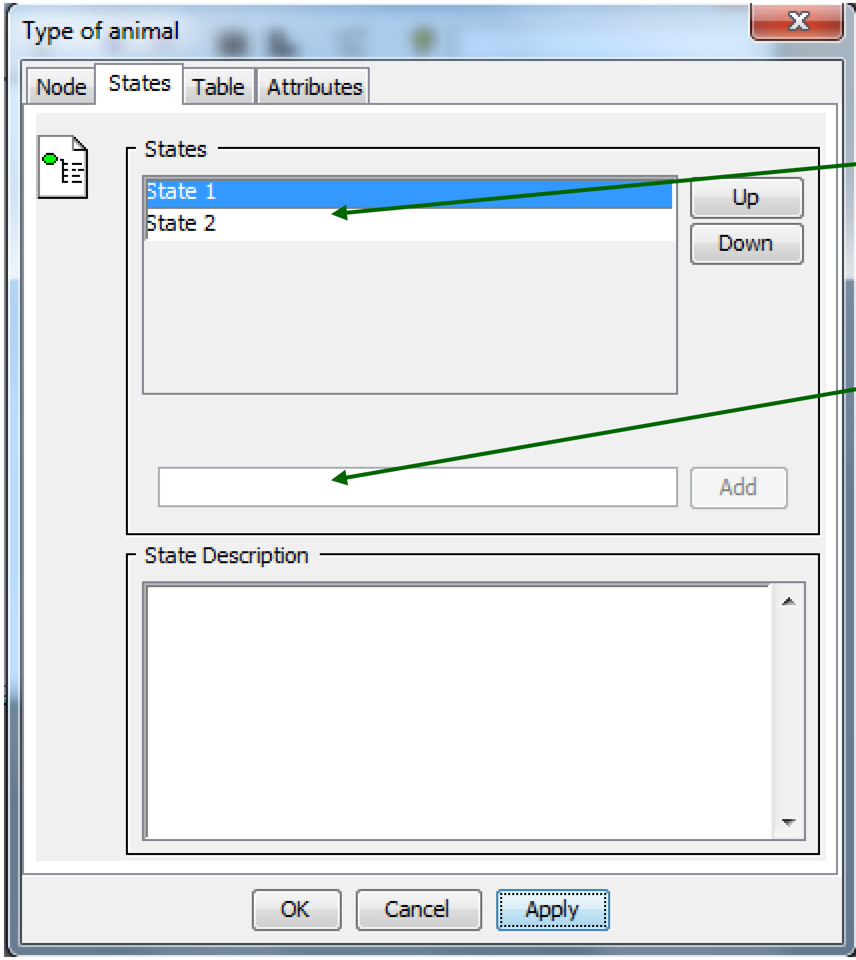
(Free format)

…for instance…

A_1 = "Willie is a cat"
A_2 = "Willie is a parrot




Type of an...
Type of animal
Node
States
Table
Attributes
Name:
A
Label:
Type of animal
Type:
Labelled
Group:
No Group
Interface:
Input
Output
Input Panel:
include in panel
Size:
Width = 80
Height = 40
Node Description
Descriptions are available on SHIFT + Right Mouse Button
OK
Cancel
Apply
Select tab "States"

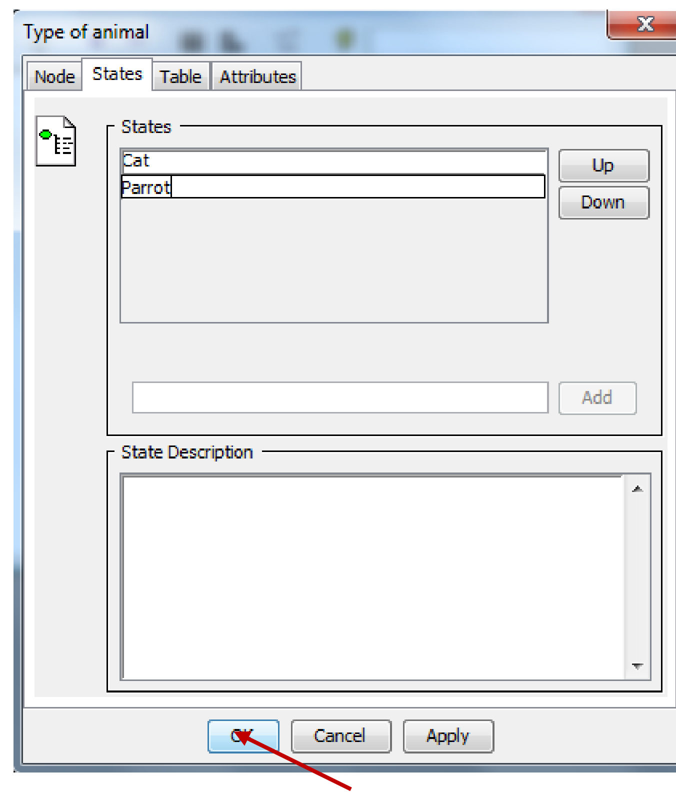


Two states per default

State names can be altered (double-click)

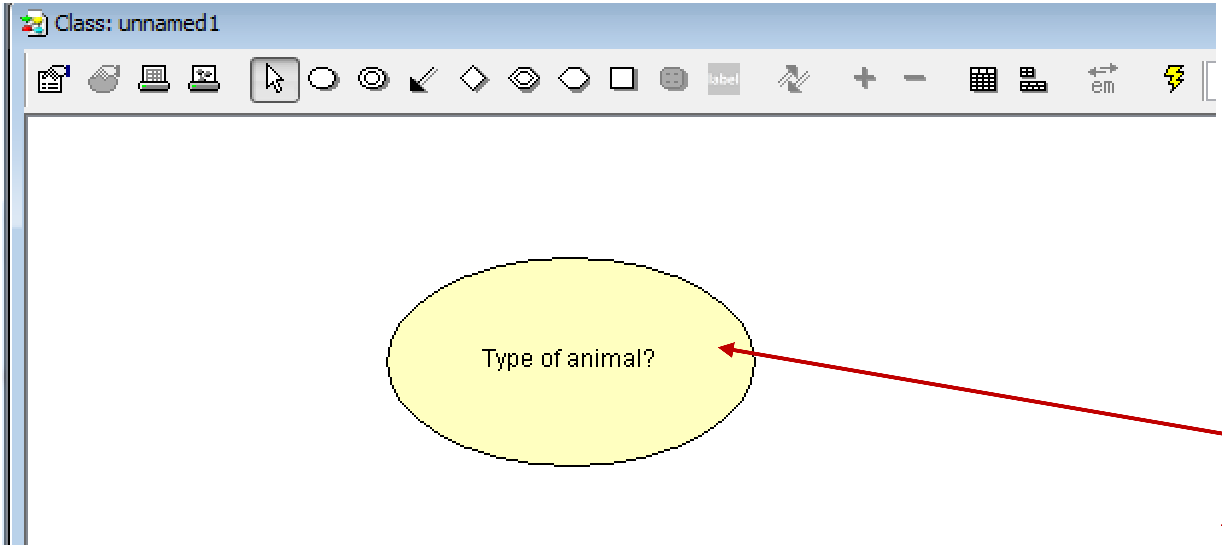
New states can be added

for instance…

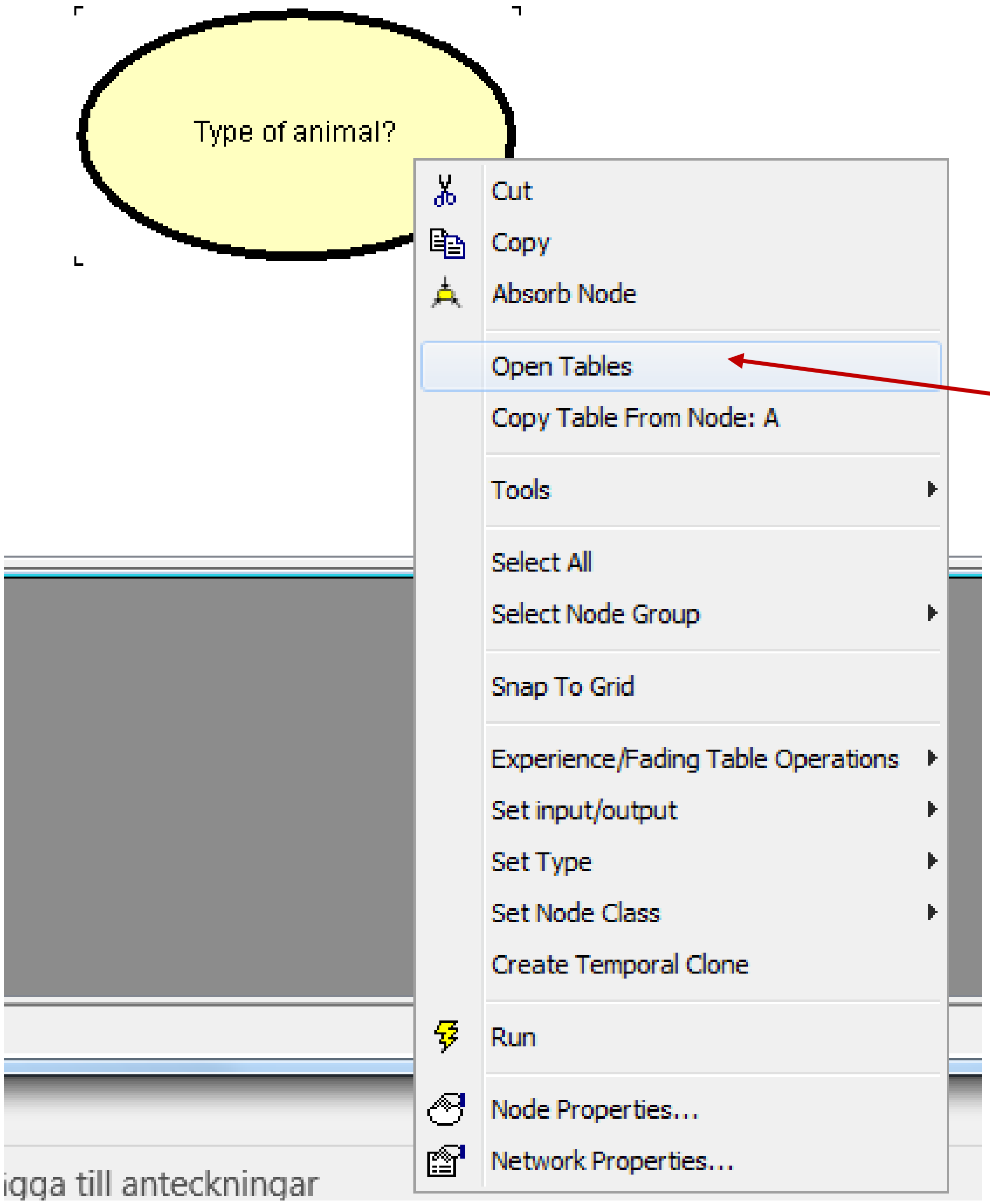


A_1 = “Willie is a cat”
A_2 = “Willie is a parrot

No more selections needed in this dialogue.

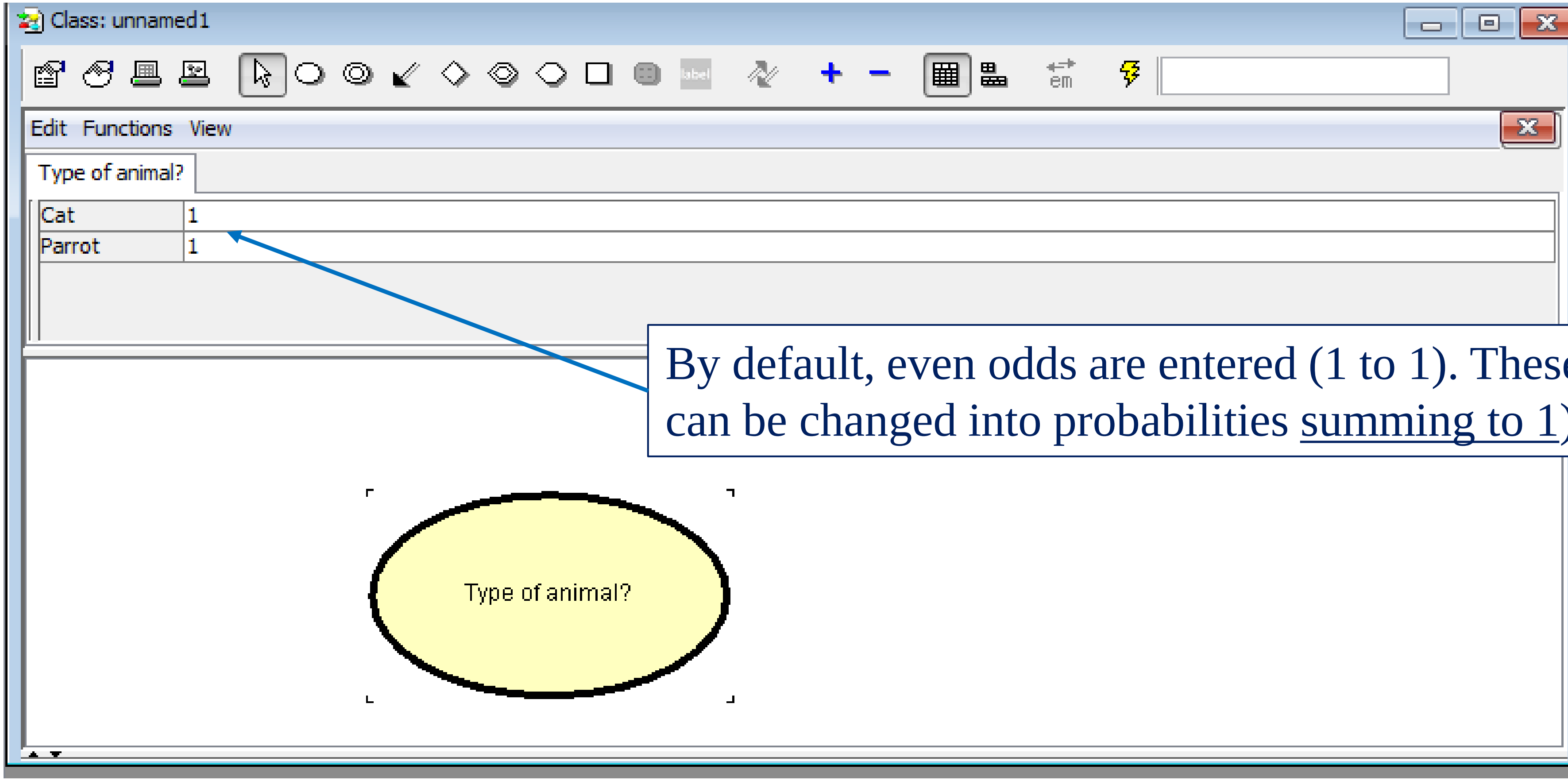


Activate by one left-click (if not active) and right-click inside to open menu

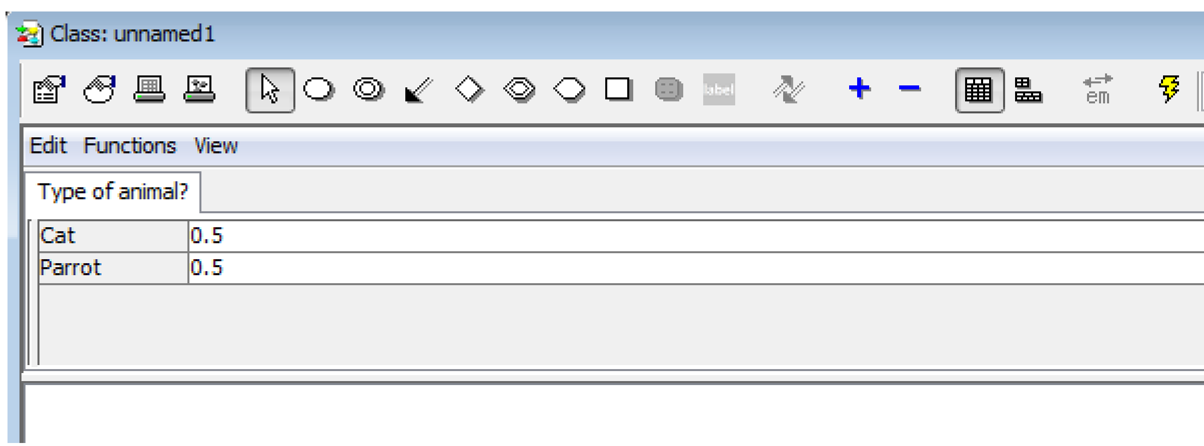


Select "Open Tables"

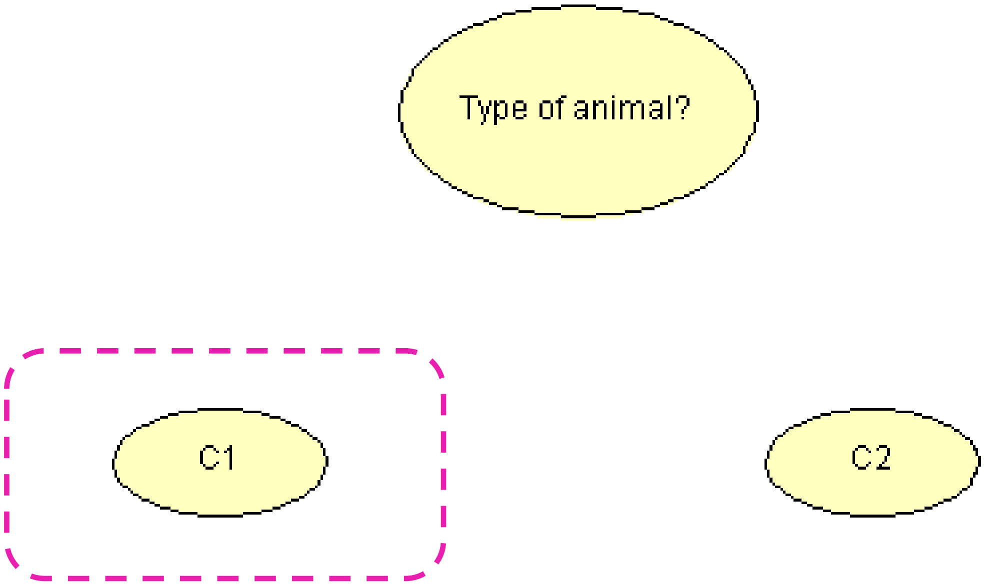
…might be followed by a warning with instructions about how a node table should be visible



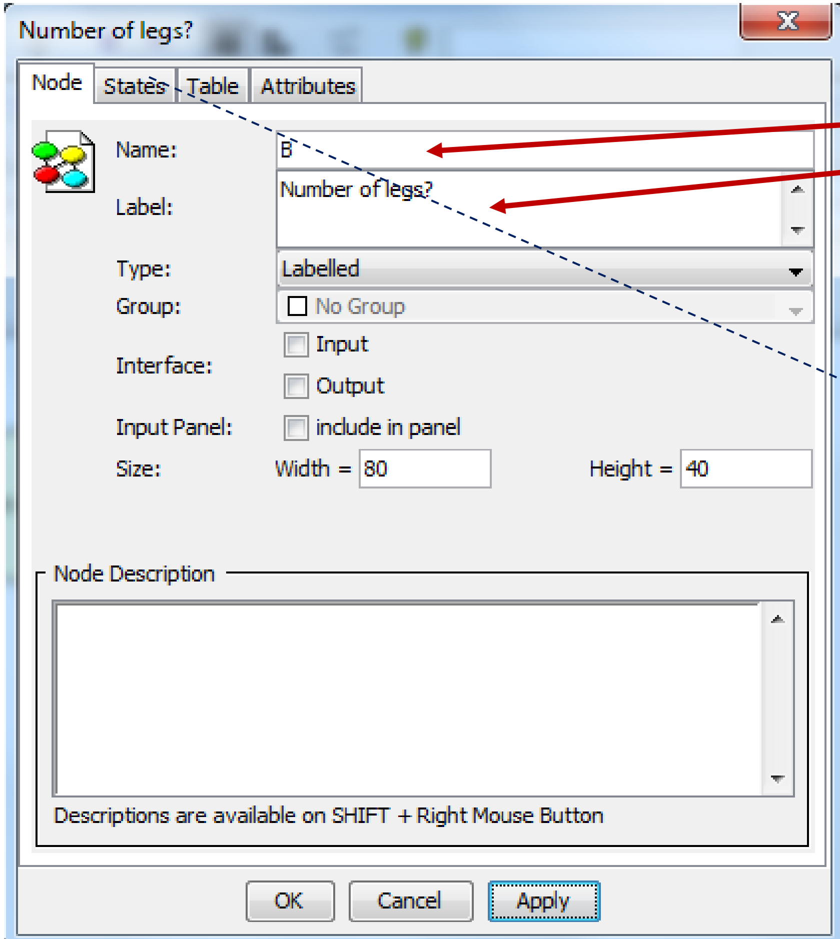
By default, even odds are entered (1 to 1). These can be changed into probabilities summing to 1).

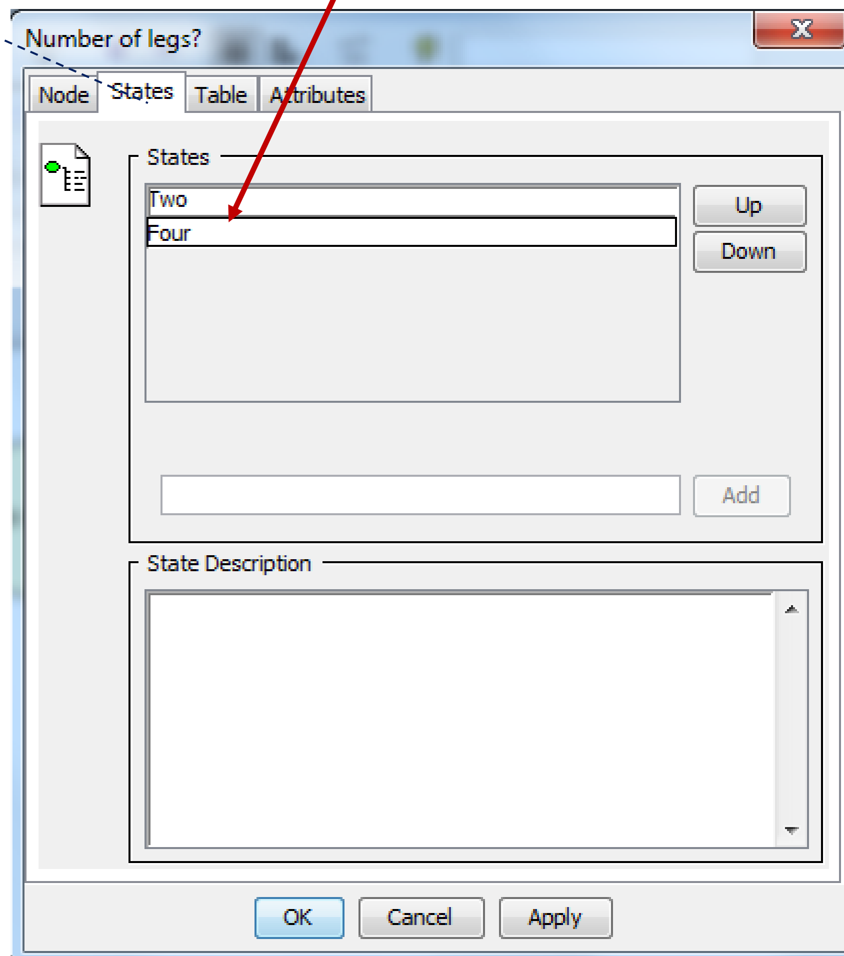


Add two more chance nodes…



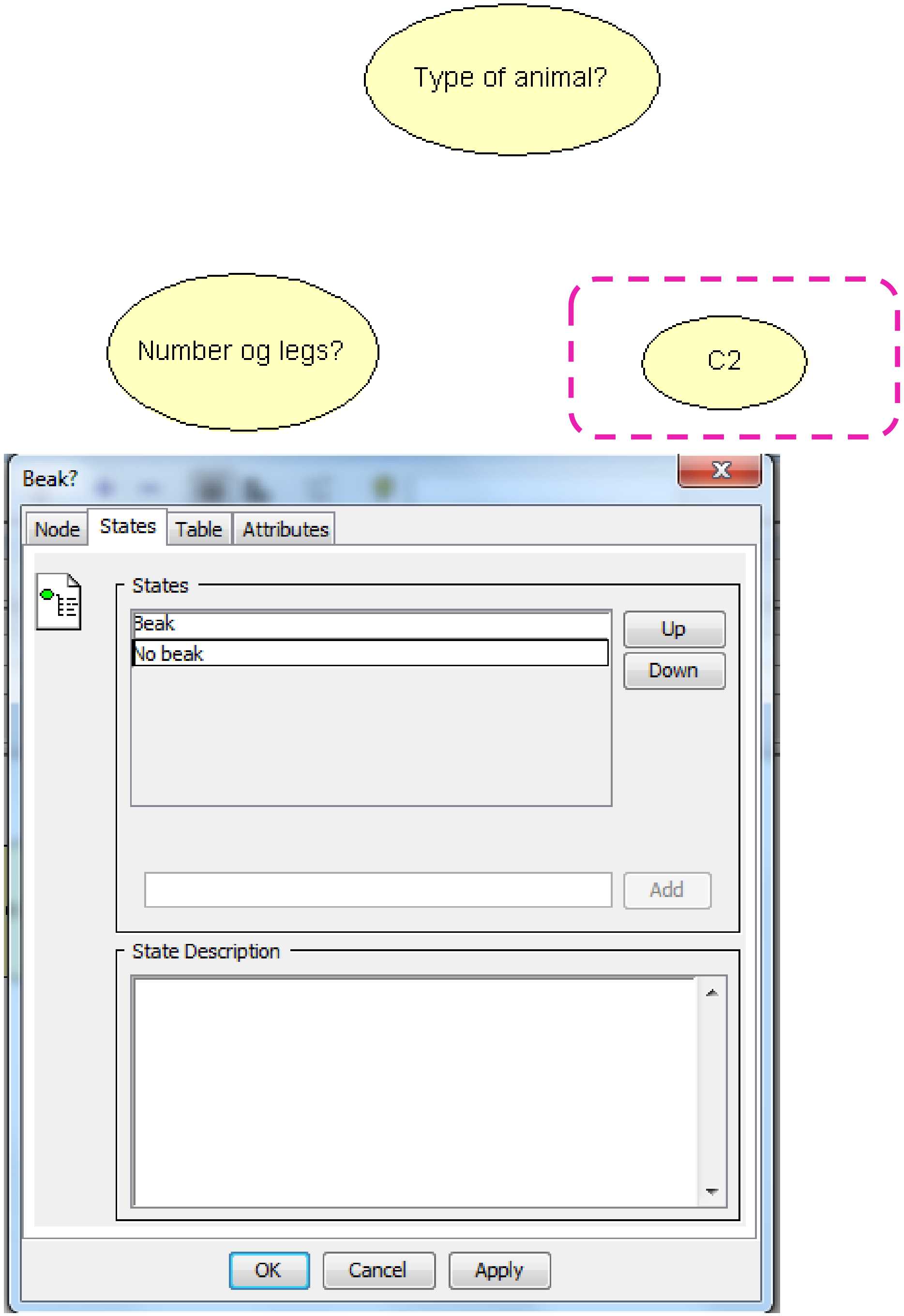
B_1 = "Willie has four legs"
B_2 = "Willie has two legs"

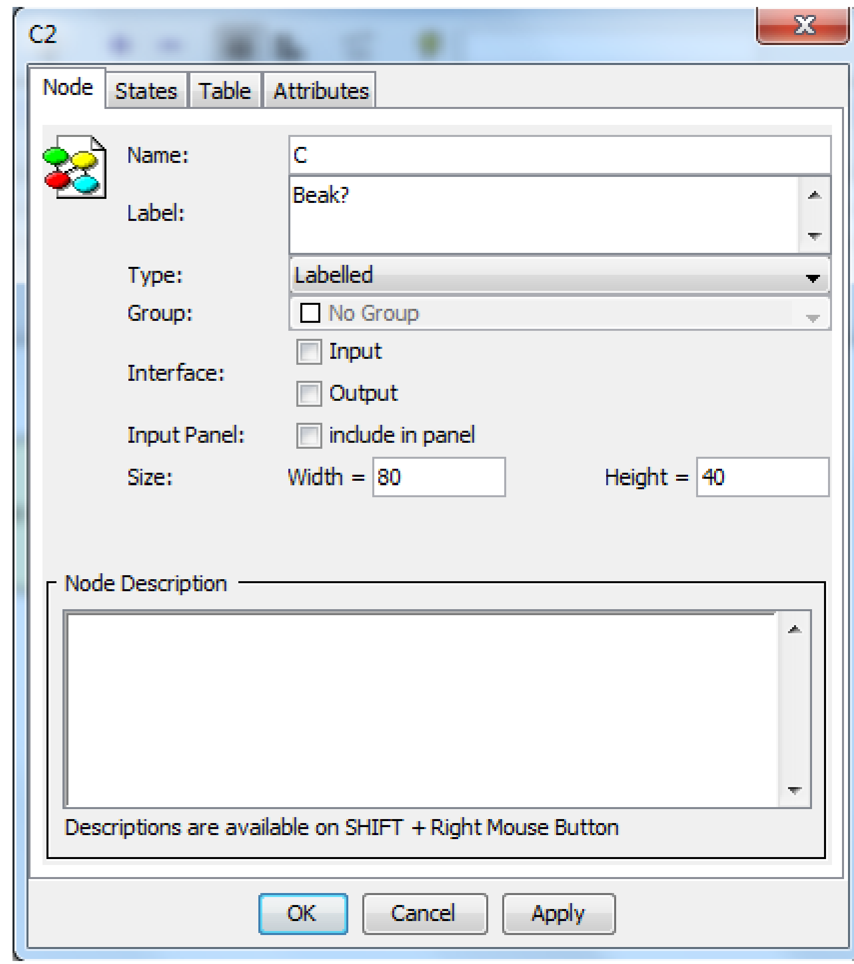




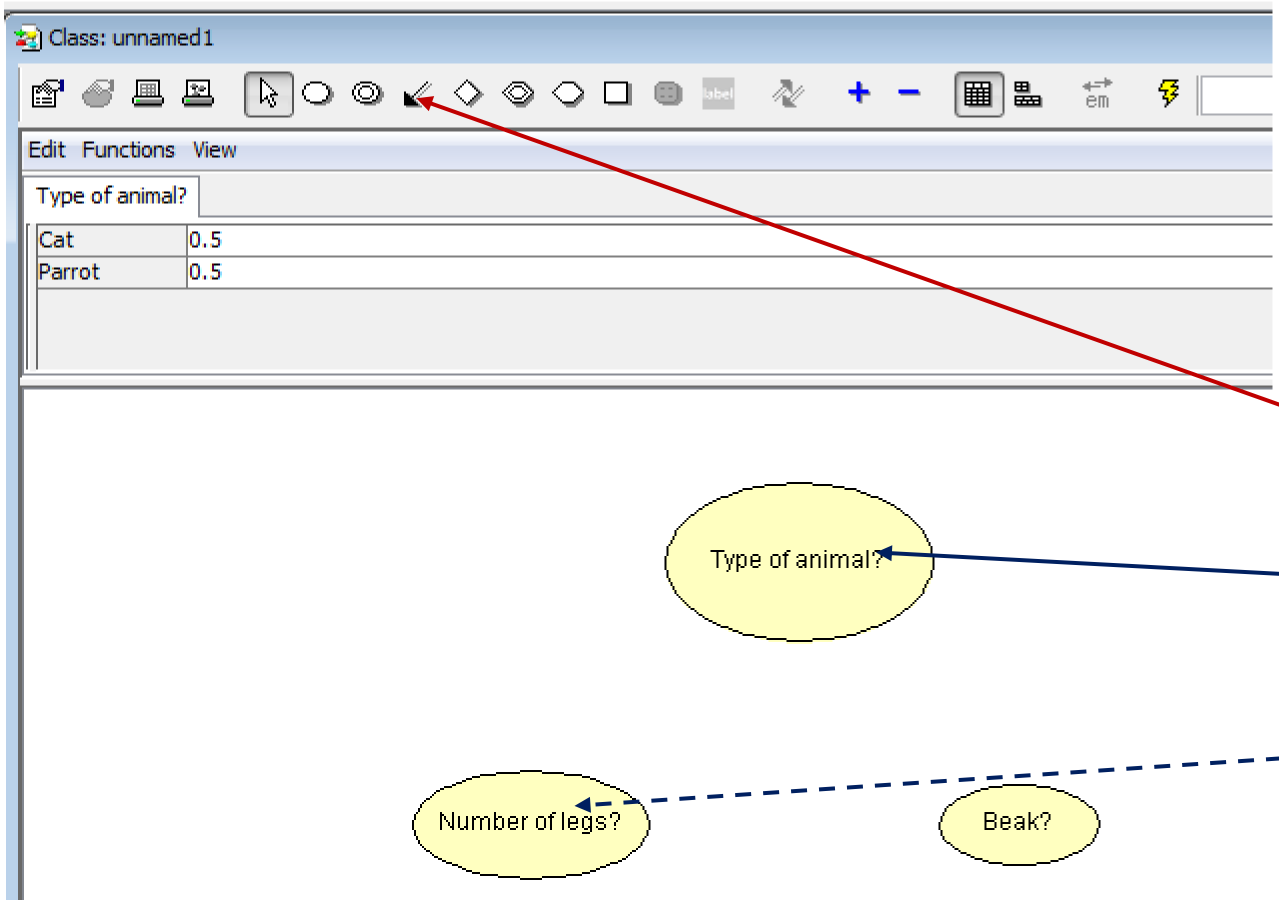
At this moment, do not bother about the probabilities.

C_1 = “Willie has a beak”
C_2 = “Willie has no beak”





Add links (edges)…

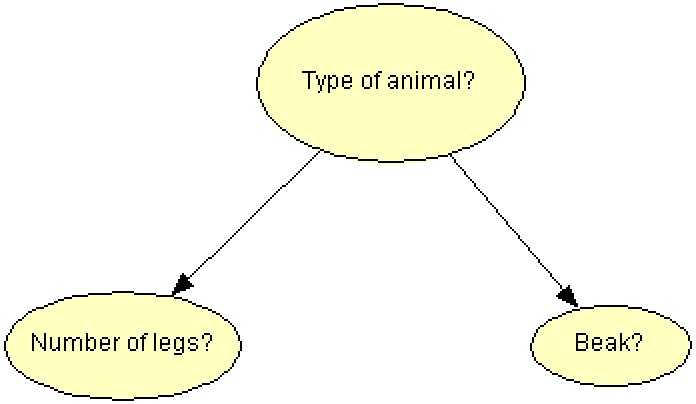


Click here

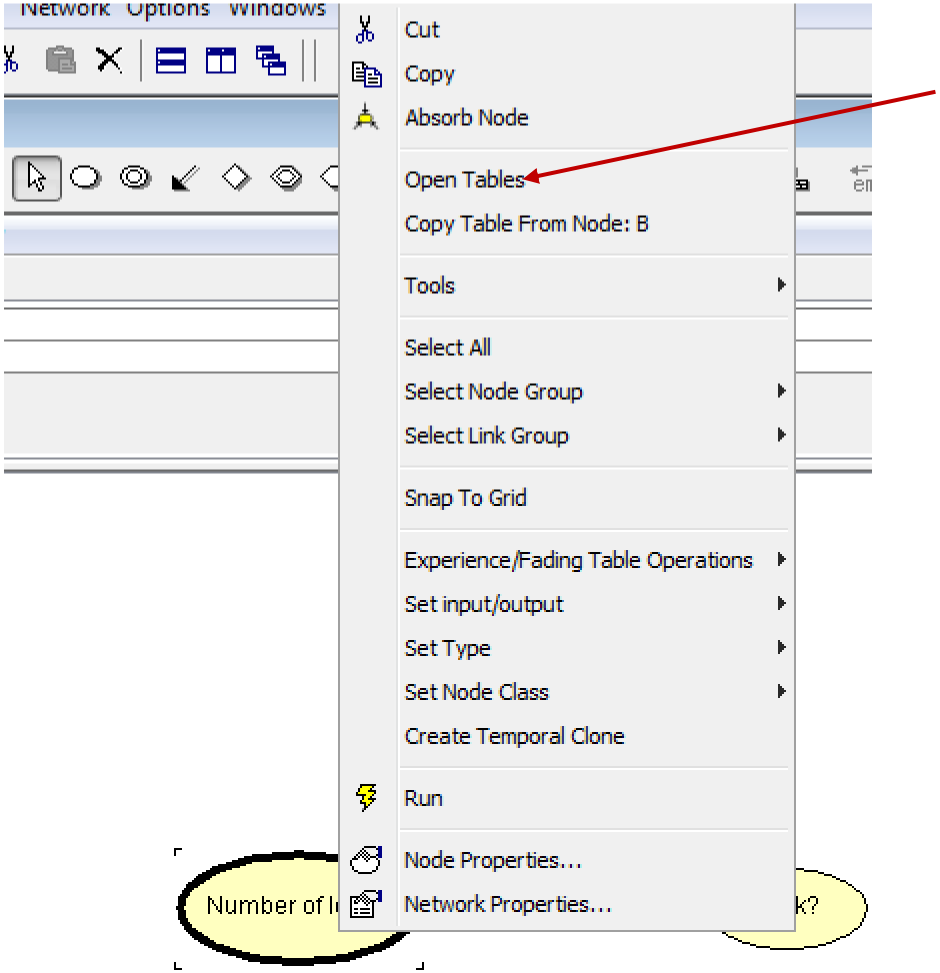
then

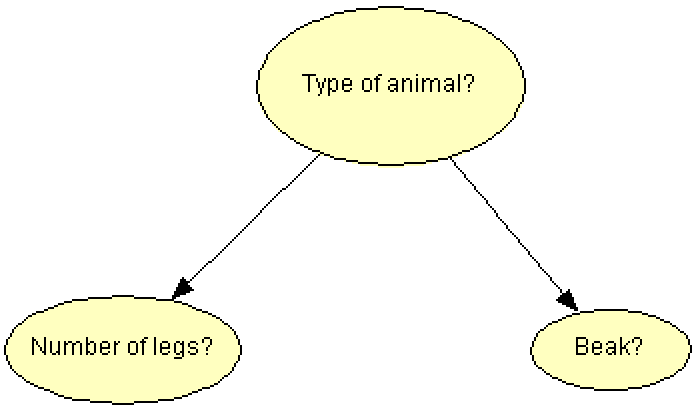
Place cursor in the <u>interior</u> of node A (“Type of animal?”), <u>drag</u> to the interior of node B (“Number of legs?”), and release.

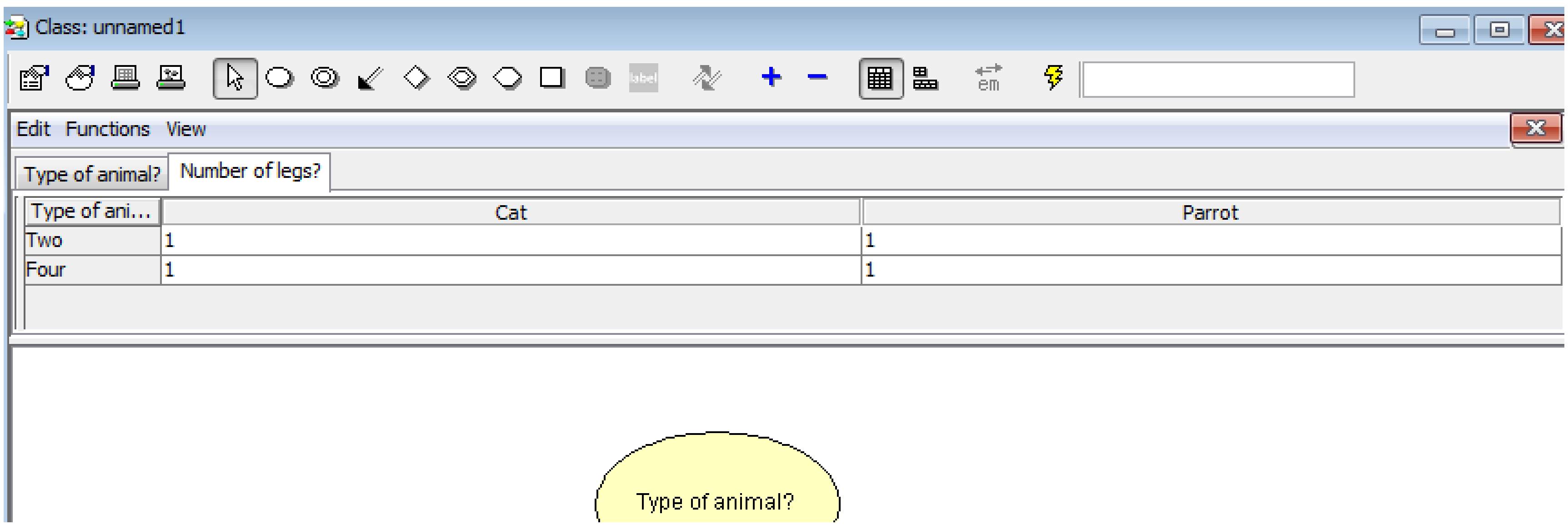
Repeat for link between A and C



Now, activate node B (Number of legs?), right-click and select Open Tables



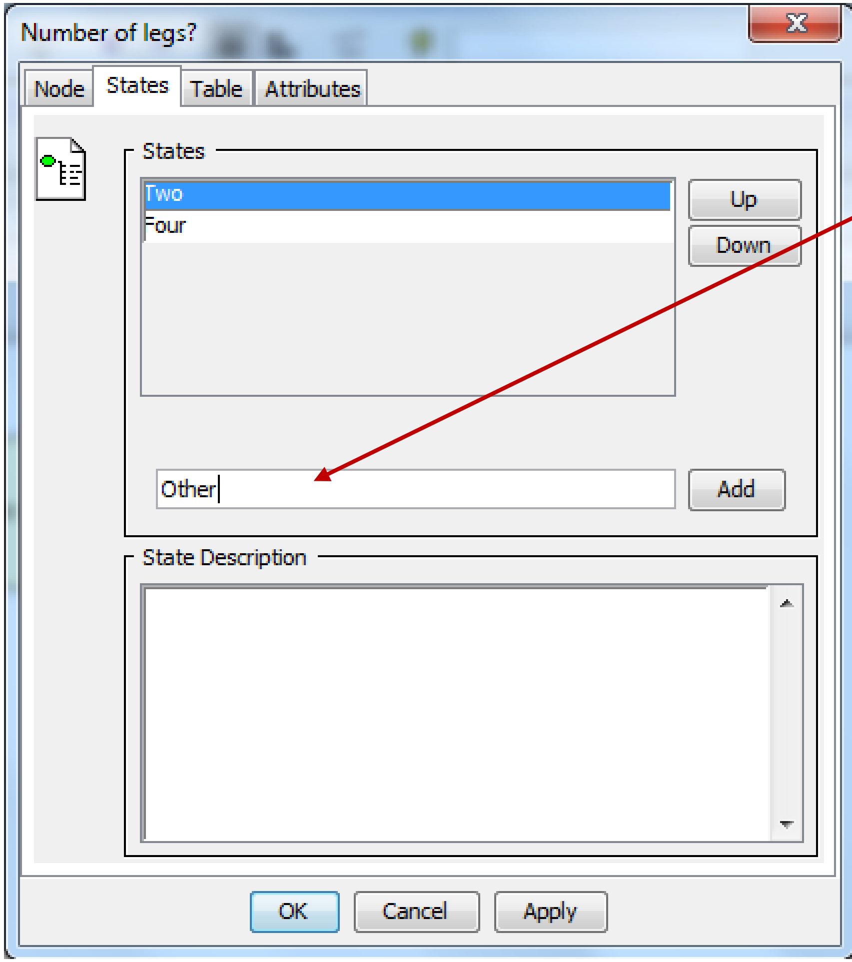




The probability table of node B now has probabilities conditional on the state of node A. The are all set to 1 per default, and will each be treated as 0.5.

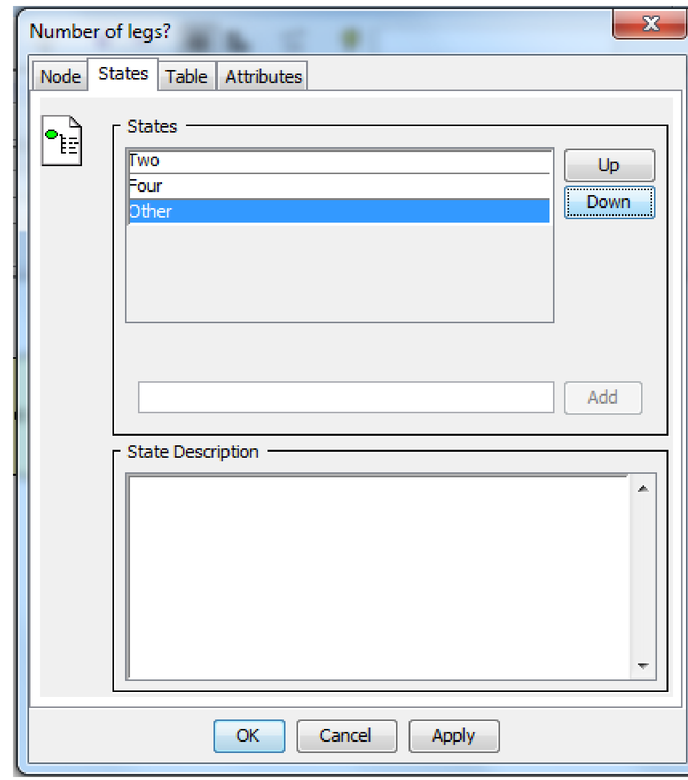
Are the states sufficiently many?

Add a state.
Double-click on node, select tab States



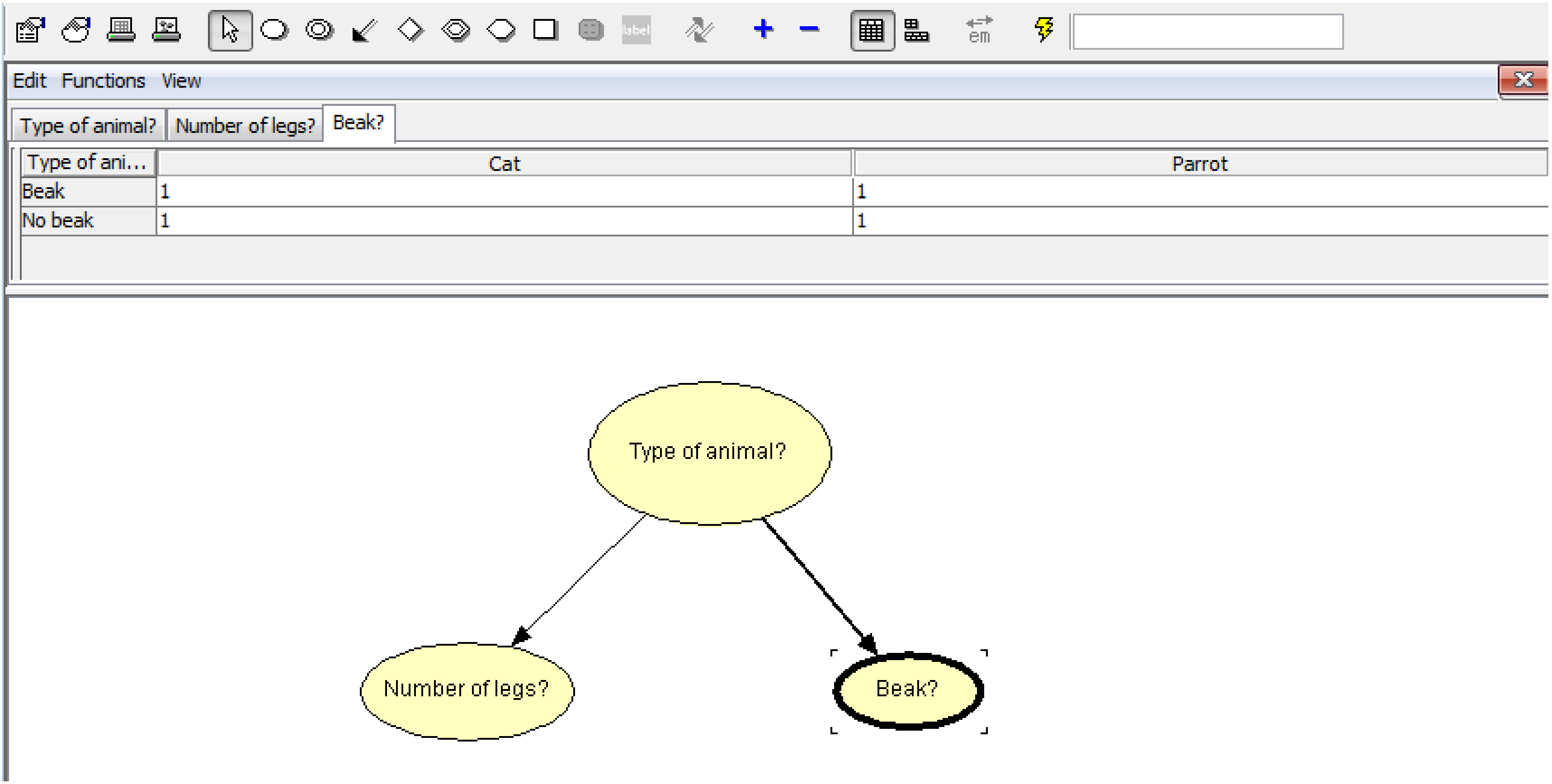
Enter name of new state here, press Add

…and a new state is added



The order of the states can be changed by pressing Up and/or Down

Now, open the table for node C (Beak?) as well.



We can now enter our probabilities into the tables.

Node C:

Edit Functions View

Type of animal? | Number of legs? | Beak?

Type of ani...	Cat	Parrot
Beak	0	1
No beak	1	0

Reasonable?

Node B:

Edit Functions View

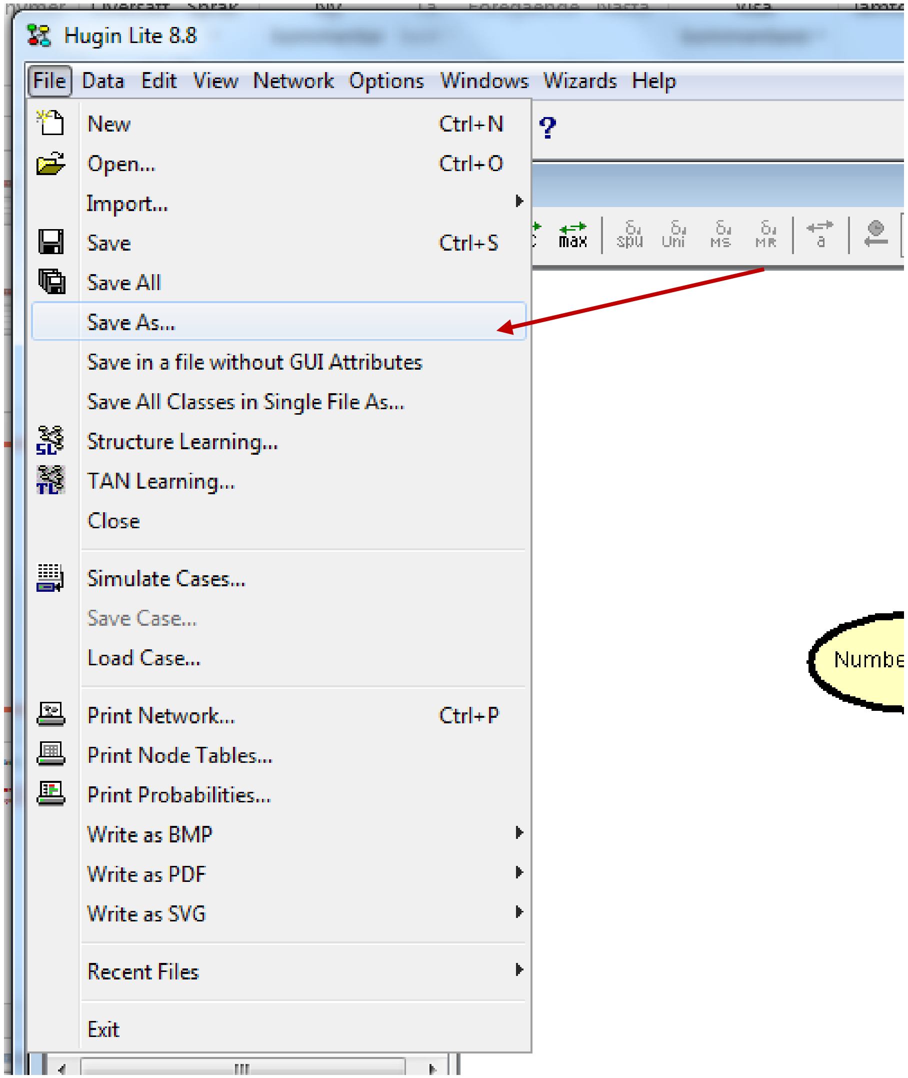
Type of animal? | Number of legs? | Beak?

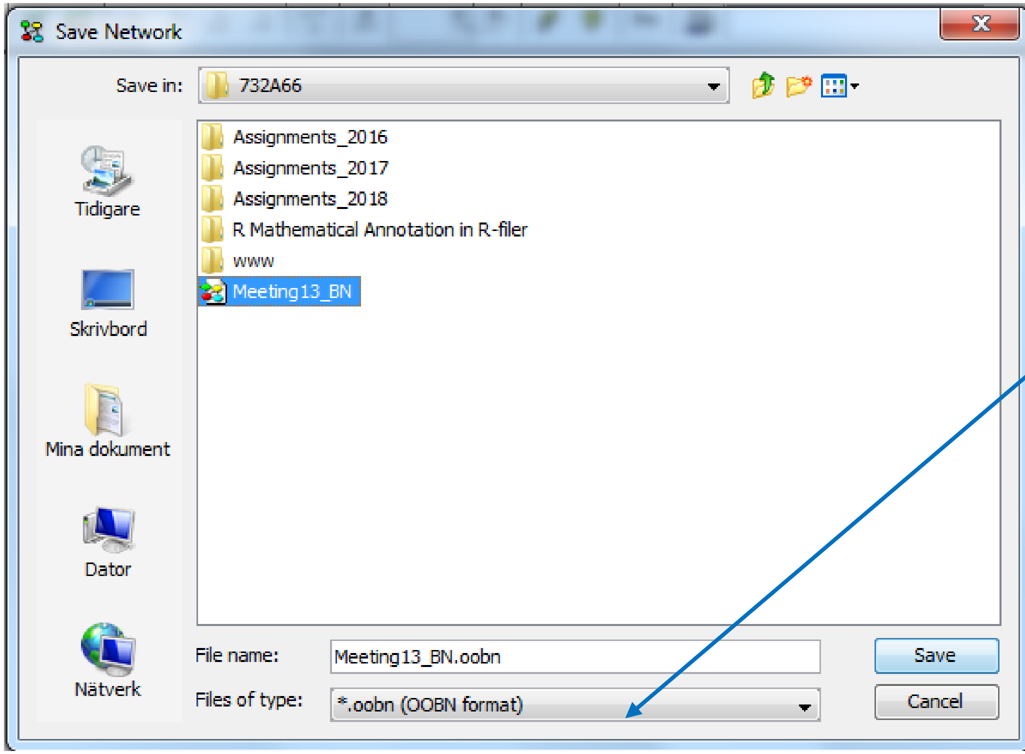
Type of ani...	Cat	Parrot
Two	0	0.9999
Four	0.9999	0
Other	0.0001	0.0001

Reasonable?

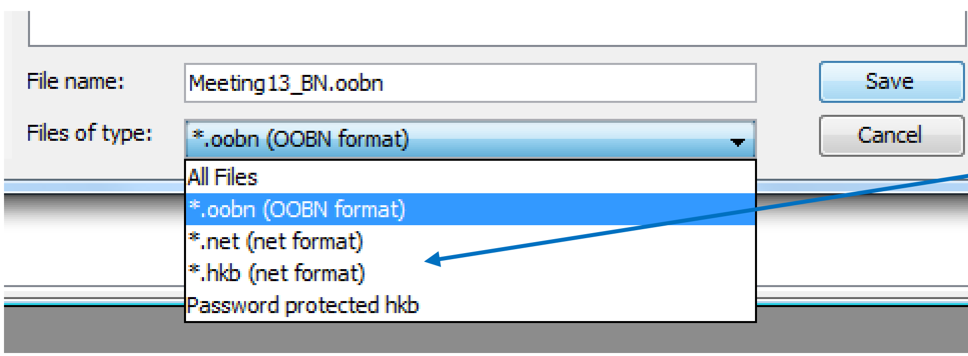
At this moment (or even earlier) it is wise to save the constructed network.

Open the File menu of the GUI and select Save As…



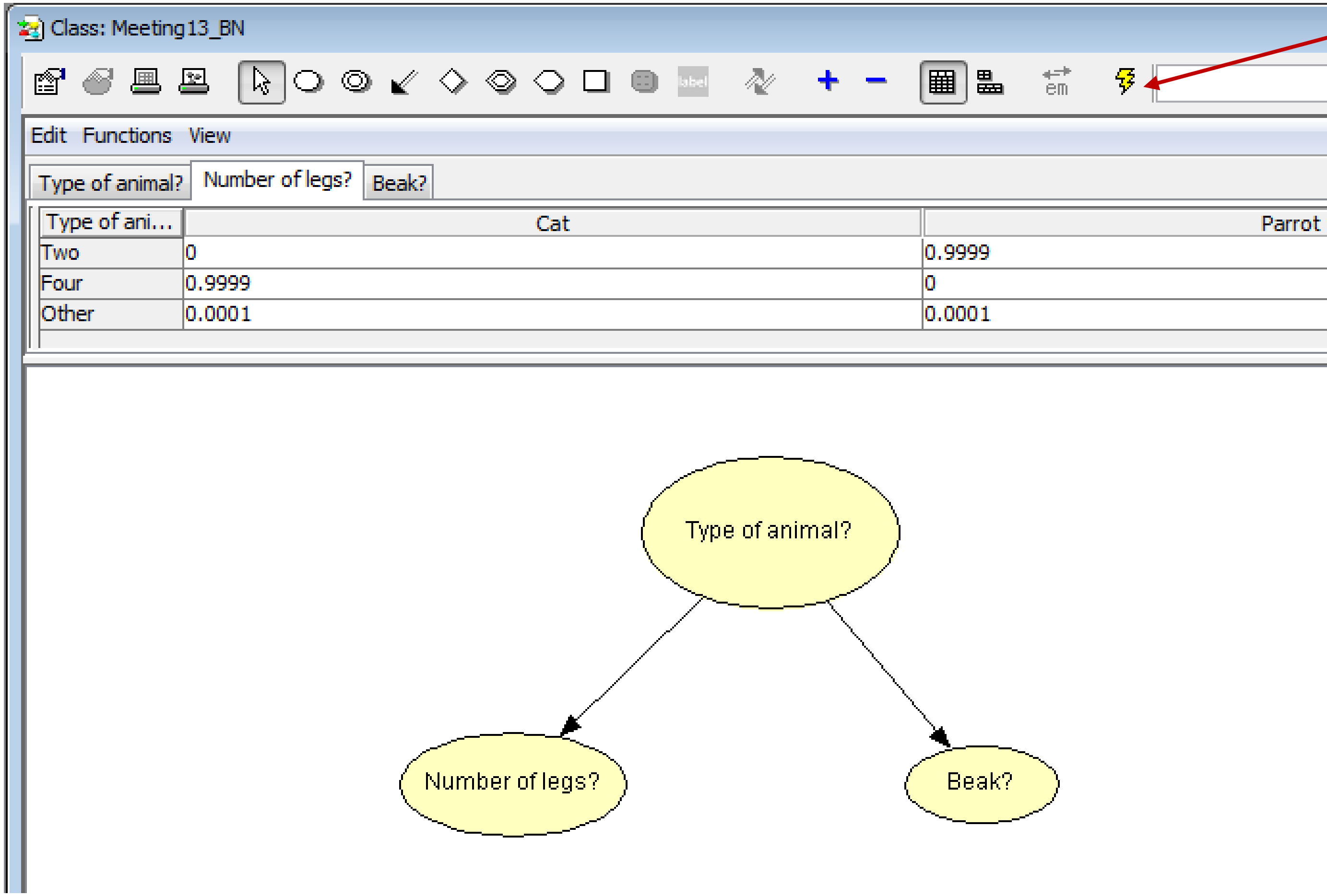


This is the Hugin format (OOBN)

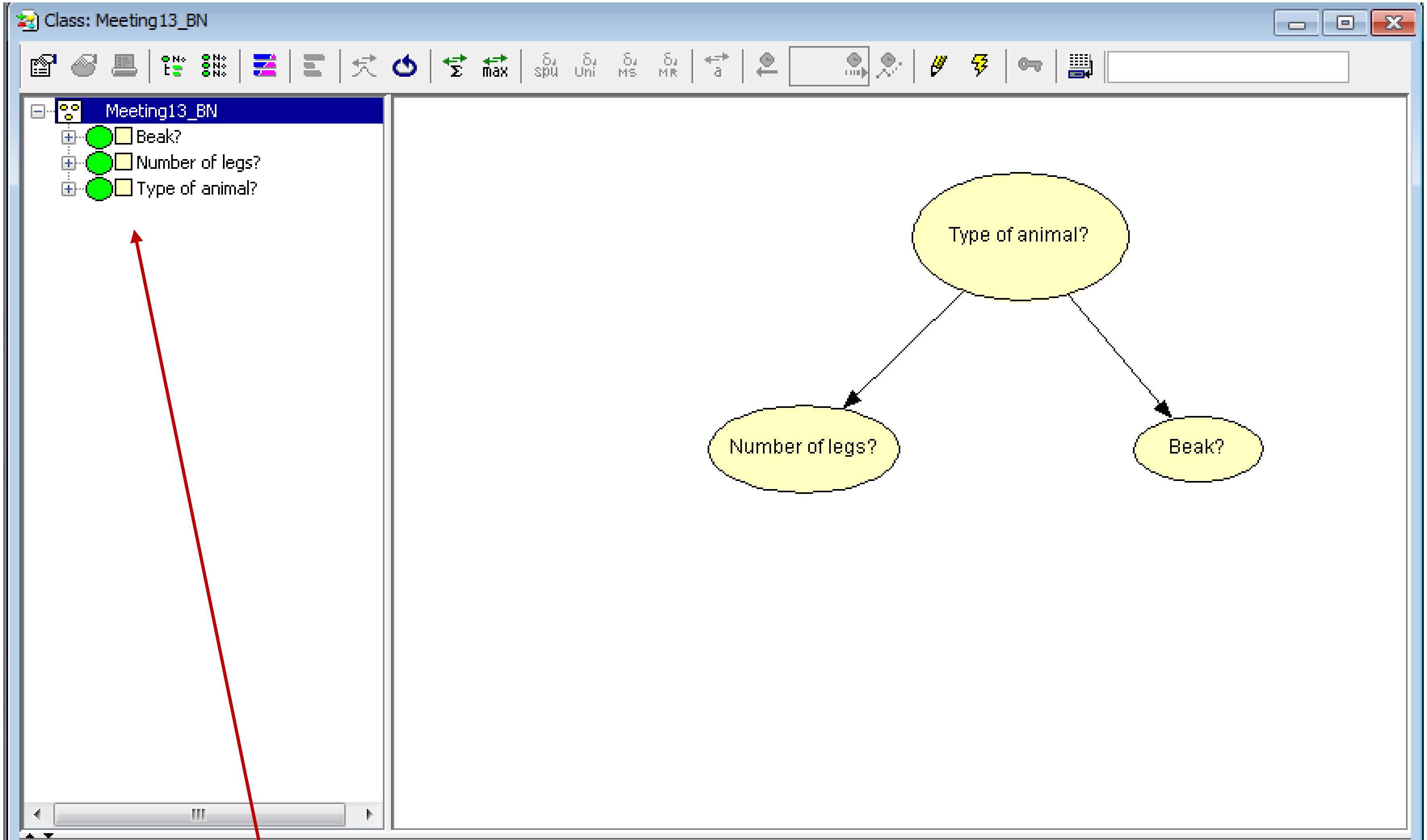


But it is also possible to use the more general NET format for compatibility with other software (like GeNIe)

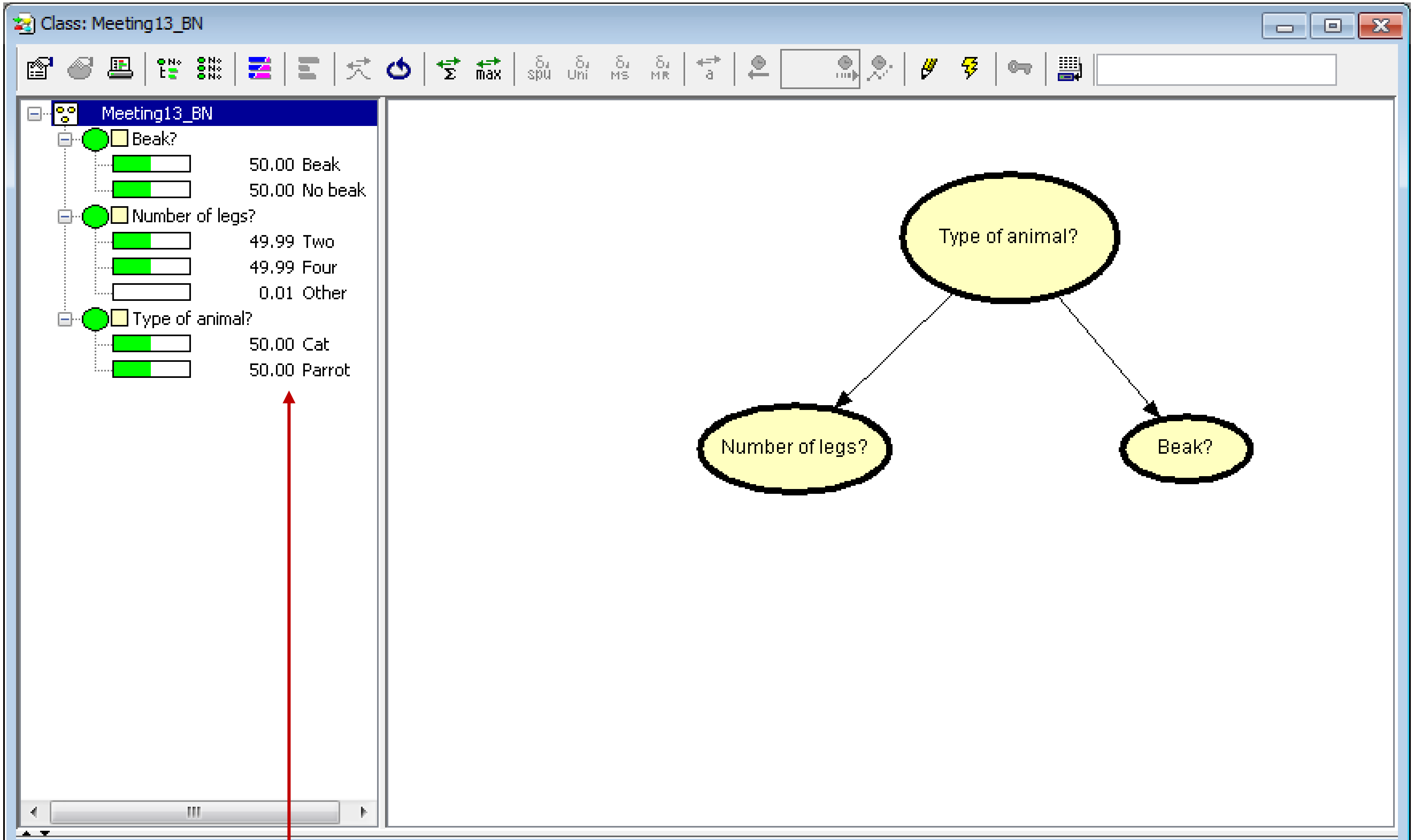
Now, the designed network should be “run”. This means putting the probabilities set into “action”



Click on the flash symbol



Expand this list



Here, we can read of the marginal probabilities (in %) of each state in each node.

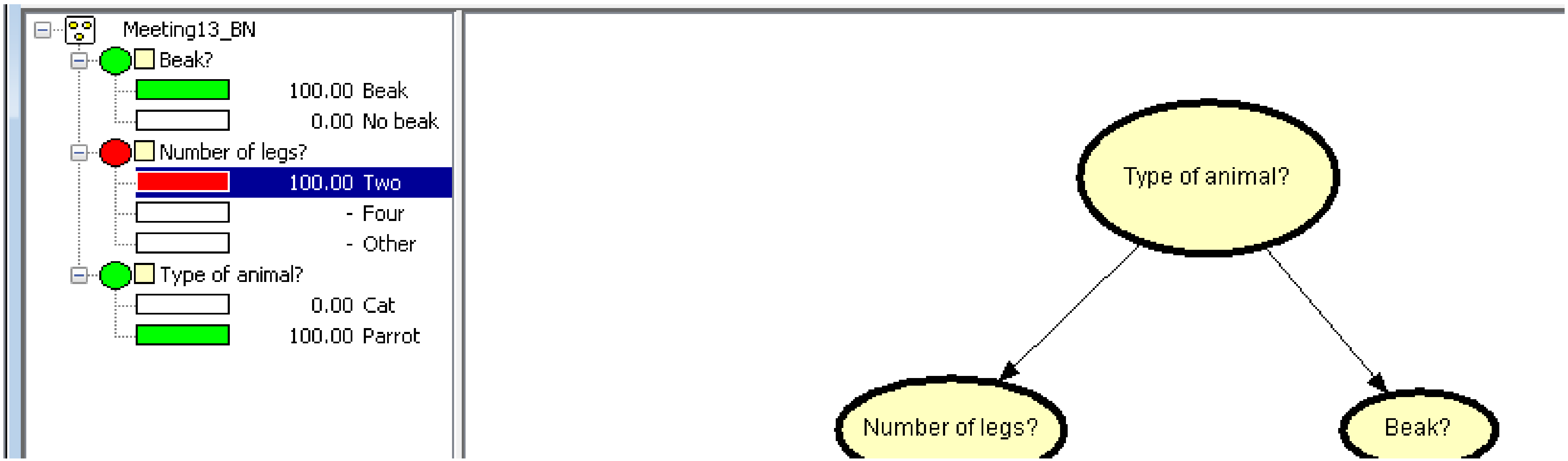
$P(\text{Beak}|I), P(\text{No Beak}|I); P(\text{Two legs}|I), P(\text{Four legs}|I), P(\text{Other}|I)$

and (as previously assigned) $P(\text{Cat}|I)$ and $P(\text{Parrot}|I)$

Entering ”evidence”

It is now possible to calculate updated (conditional) probabilities given a particular state in one or several of the nodes. This is called ” to instantiate” a node to the state of interest and is done in the software by double-clicking on that state.

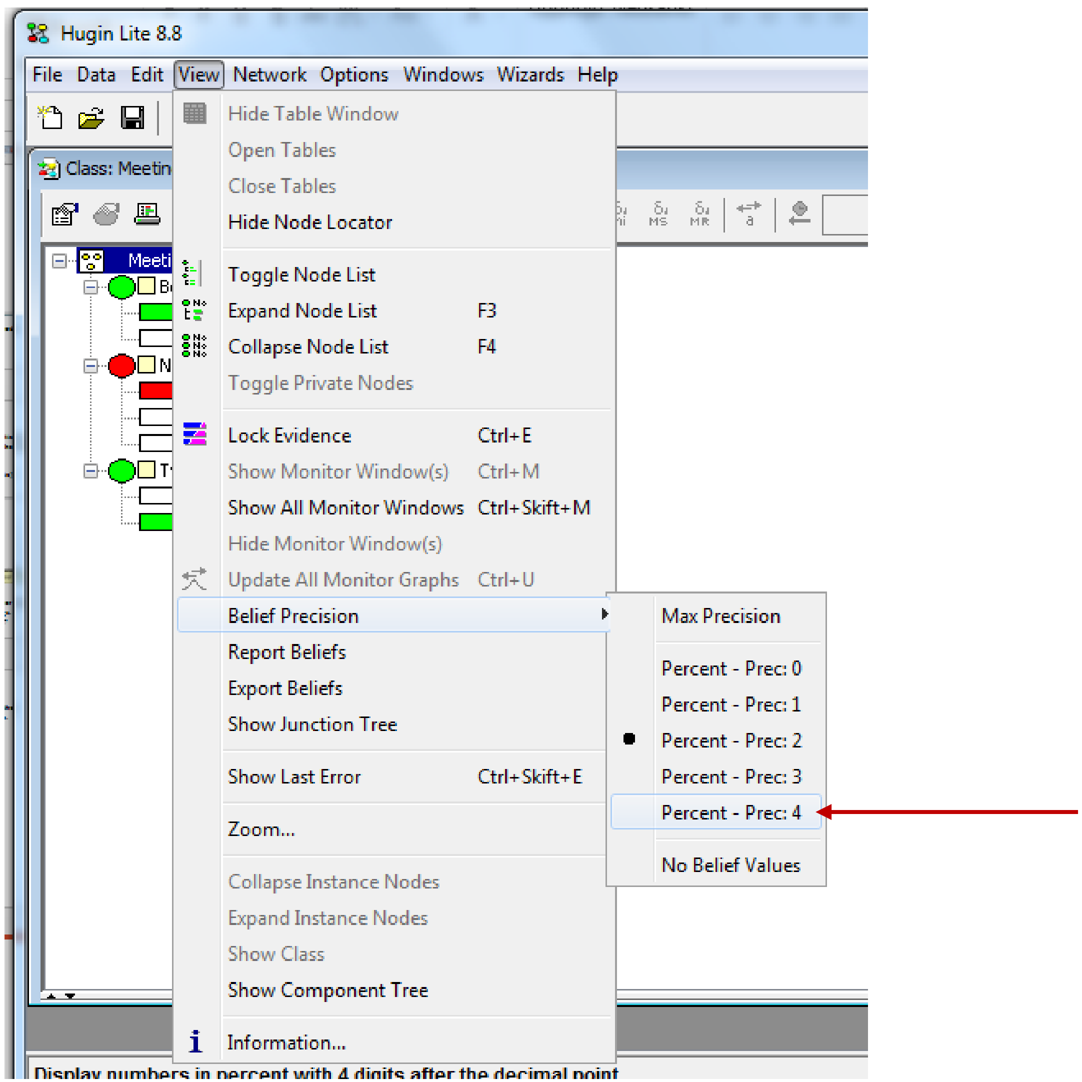
As an example, suppose that we obtain the information that the animal (Willie) has two legs. Then, we double-click on that state in the list.

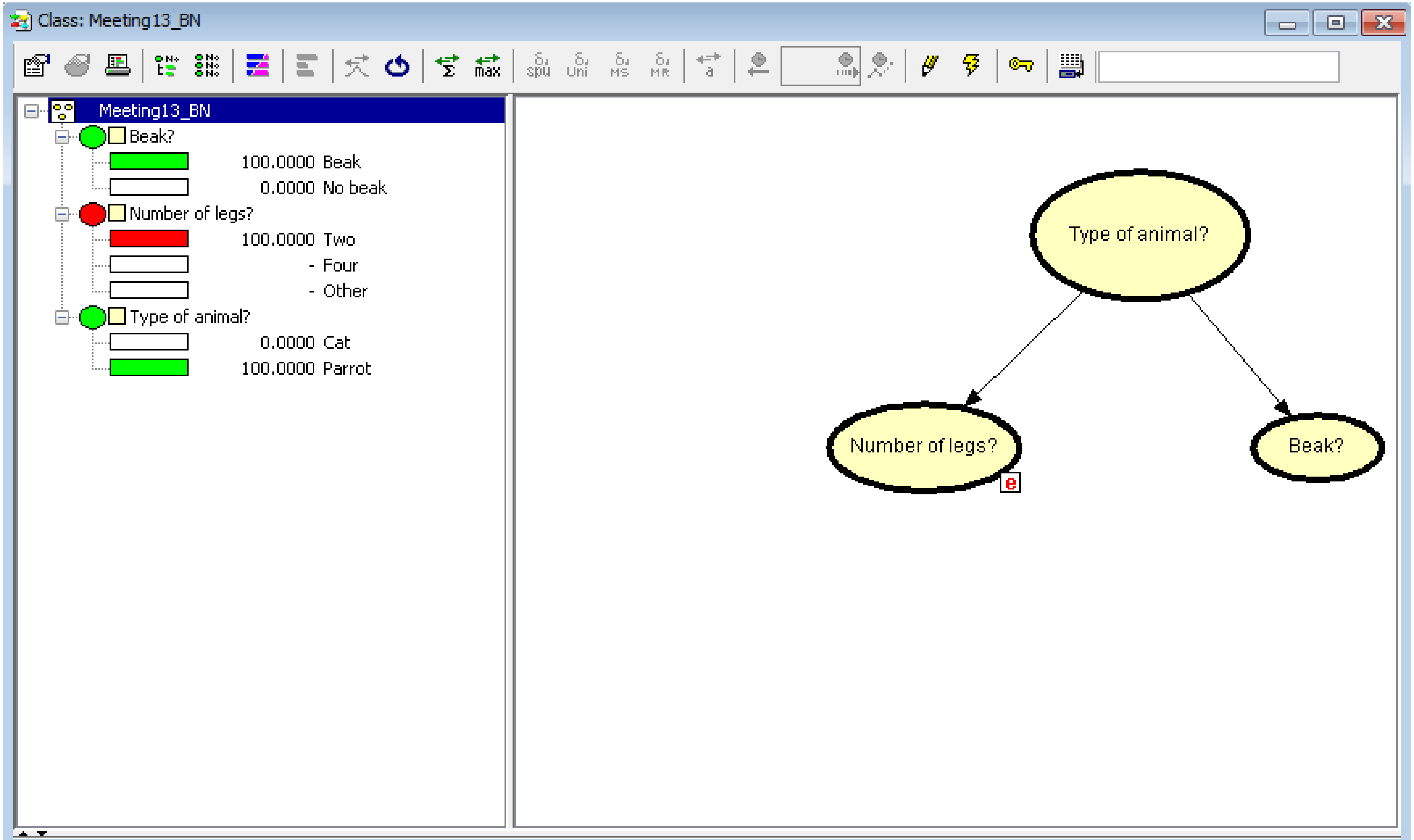


The bar colour of this state changes into read and its value to 100 %. The other bars also change values, i.e. the conditional probabilities given Willie has four legs. However, it seems they are either 100.00% or 0.00%. Could that be correct?

The precision used for displaying the numbers can be changed.

From the GUI menu select View, from the list select Belief Precision and in the following list select Percent – Prec: 4. *Other choices can of course also be made*.





Well, we have now the values 100.0000% and 0.0000% respectively. Trying Max Precision would not give more information.

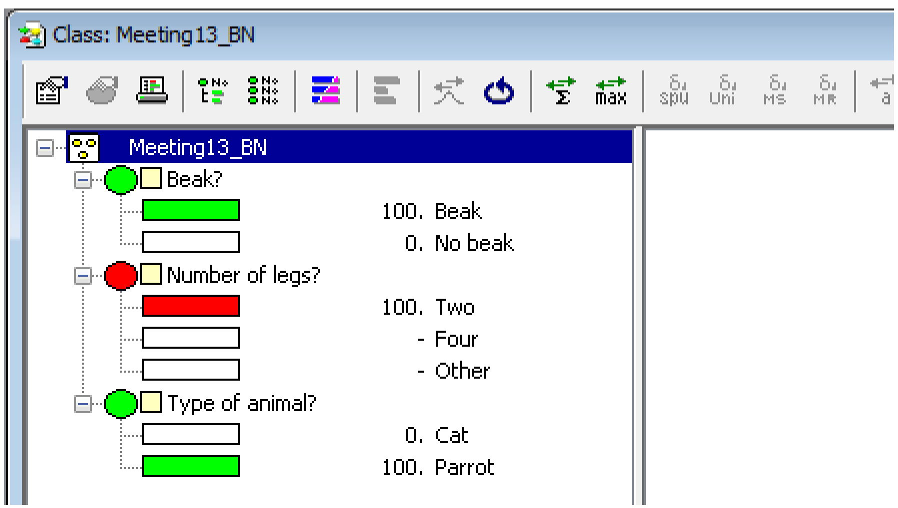
What probabilities are we computing here? Consider for instance the updated probability for A_2, i.e. ”Willie is a parrot”.

This probability is

$$P(A_2|B_1) = \frac{P(B_1|A_2) \cdot P(A_2)}{P(B_1|A_1) \cdot P(A_1) + P(B_1|A_2) \cdot P(A_2)} = \frac{0.9999 \cdot 0.5}{0 \cdot 0.5 + 0.9999 \cdot 0.5} = 1$$

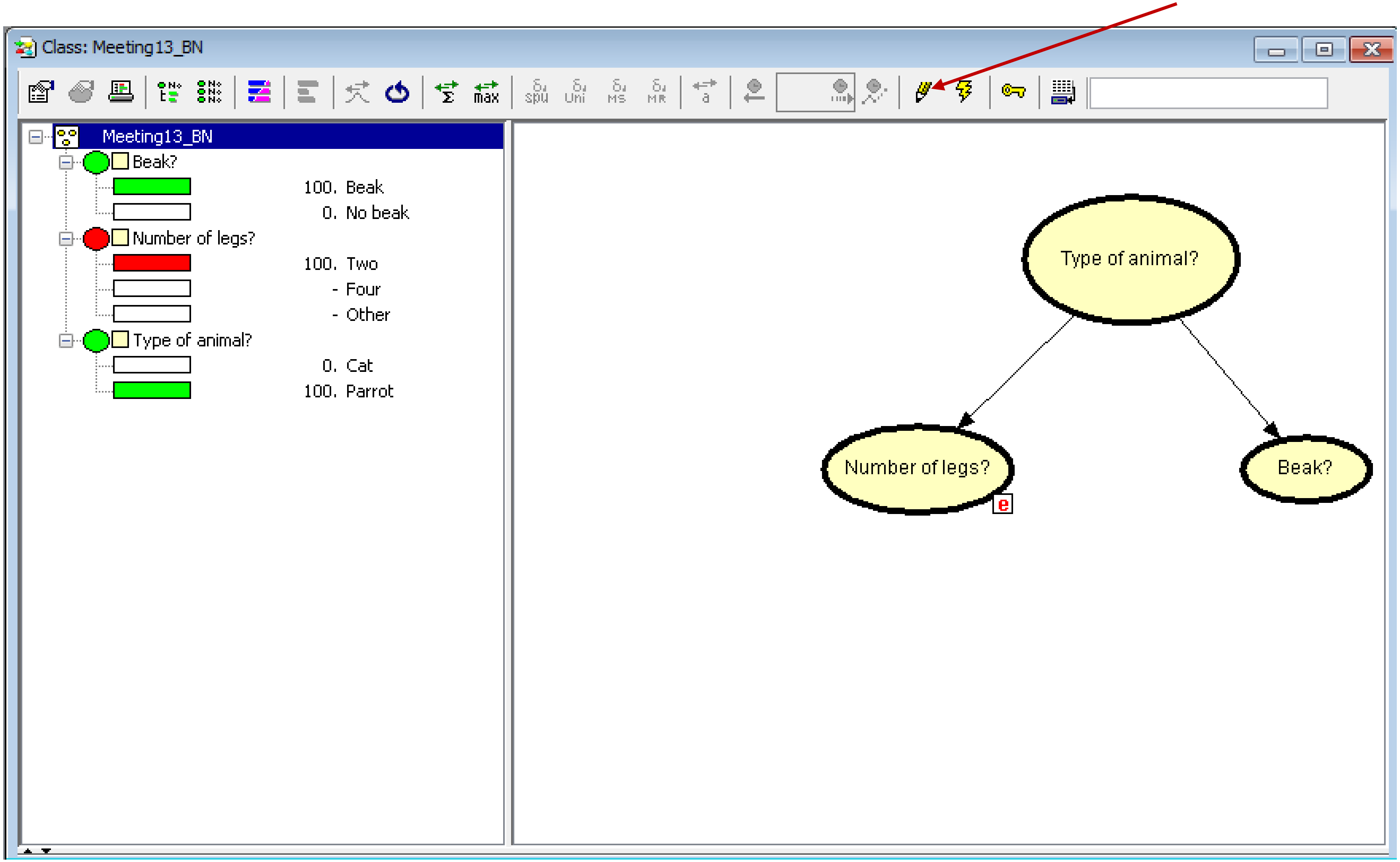
Hence, the probability is (maybe not so unexpected) exactly 100%

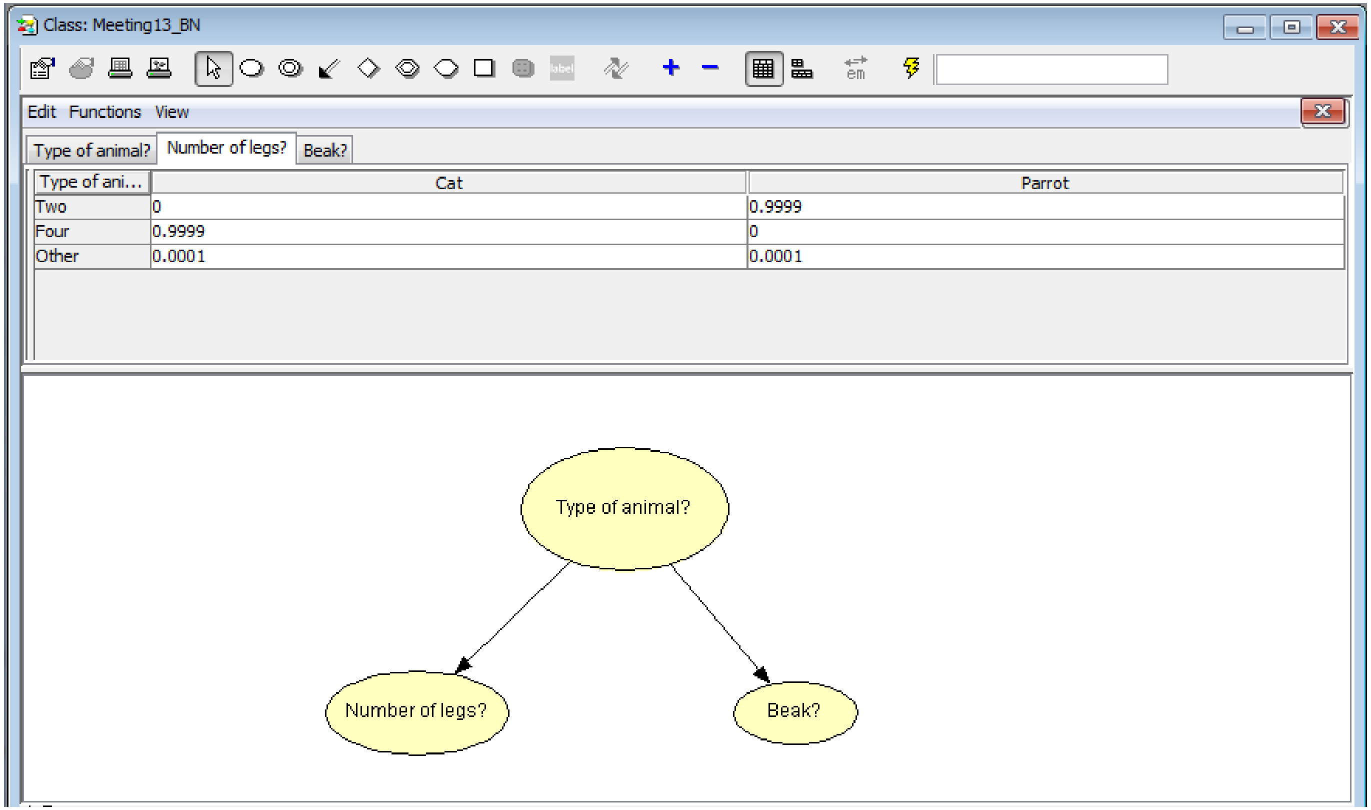
This can also be seen by setting Belief Precision to Max precision



The absence of displayed decimals indicate that the values displayed are exactly 100 and 0 respectively.

If we want to edit the network, e.g. adding nodes and/or changing assigned probabilities, we can return to Edit mode by clicking on the pencil icon.

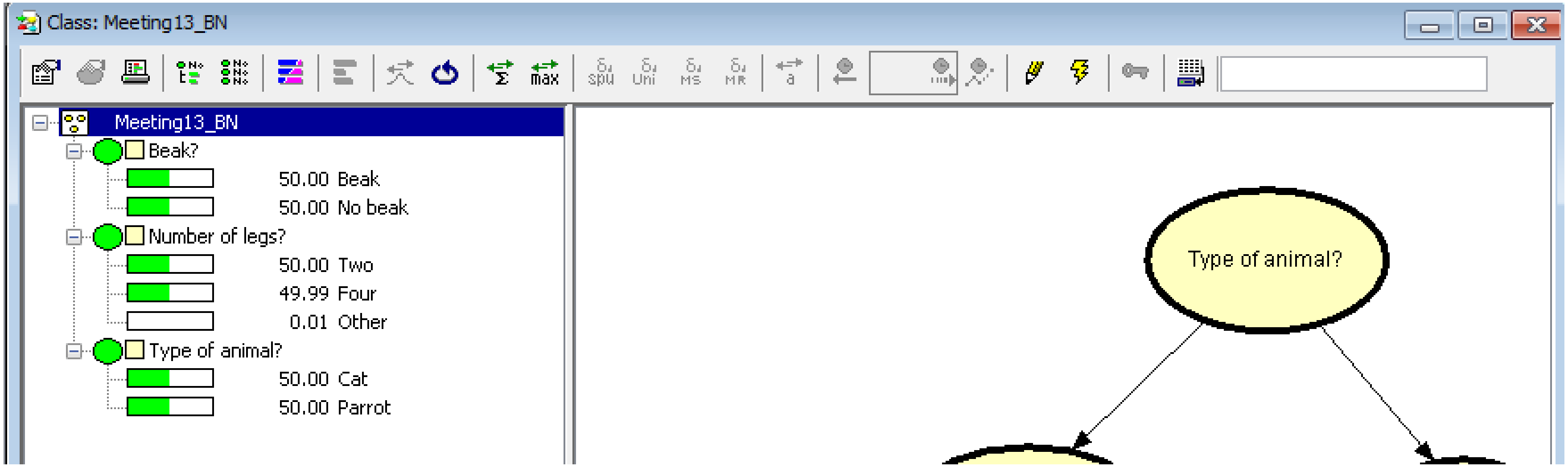




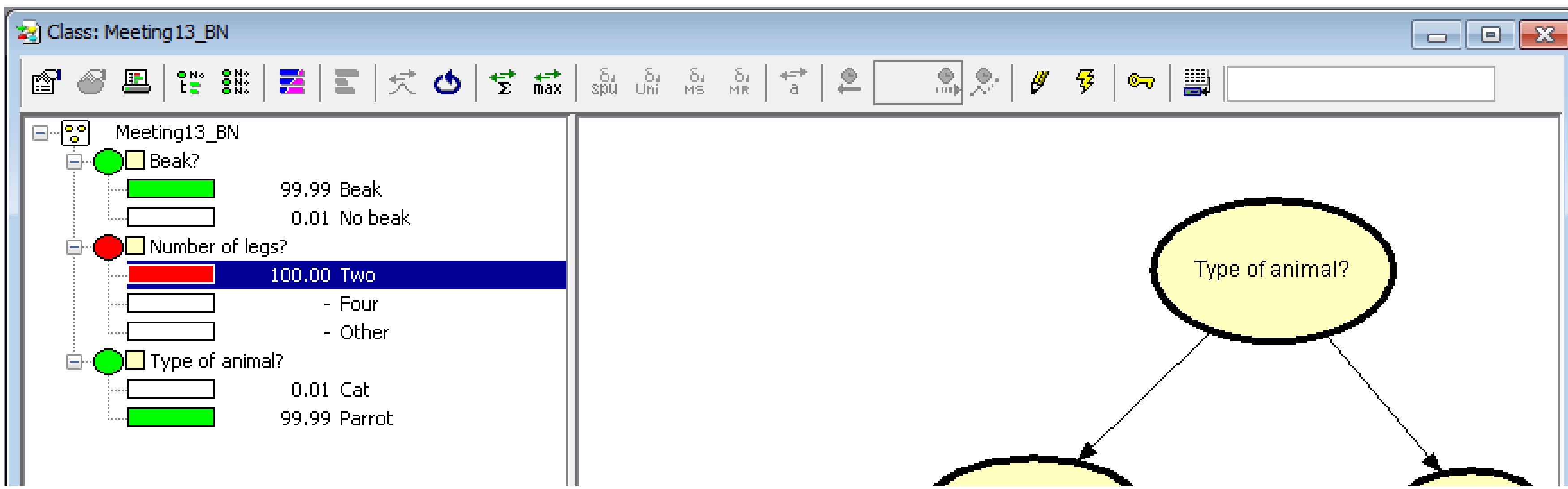
Now, change the conditional distribution of the number of legs given the type of animal is a cat to 0.0001, 0.9998 and 0.0001 respectively, and run network again.

Type of animal? | Number of legs? | Beak?

Type of ani...	Cat	
Two	~~0~~	0.0001
Four	~~0.9999~~	0.9998
Other	0.0001	0.0001



Again, instantiate state ”Two” in node Number of legs?



…and we can see that the updated probability for Willie being a parrot is now 0.9999.

Another example Who smashed the window?

A window (pane) was smashed and a person, Mr G is suspected for having done it. On Mr G's pullover 8 glass fragments were recovered, they all matched the (pane of) the smashed window.

Let

H be a random variable with states H_1= "Mr G smashed the window" and H_2: "Someone (or something) else smashed the window".

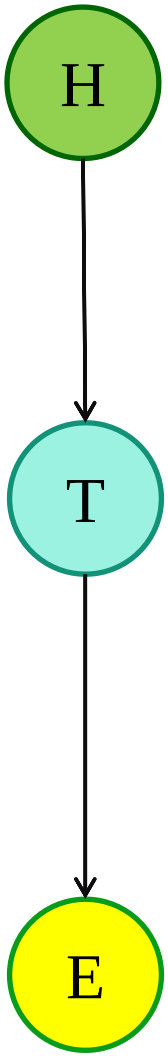
T be a random variable for which the state is the number of fragments transferred to Mr G's pullover when the window was smashed. Note that if Mr G's pullover was not sufficiently near the window when it was smashed, then $T = 0$.

E be a random variable for which the state is the number of fragments that could be (and were) recovered from Mr G's pullover. Note that E is not equal to T since (i) it cannot be assumed that all fragments transferred to Mr G's pullover persisted and (ii) were detectable when analysing it.

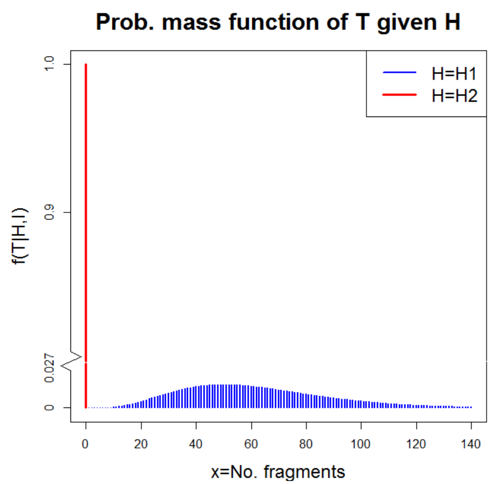
Serial *connection*

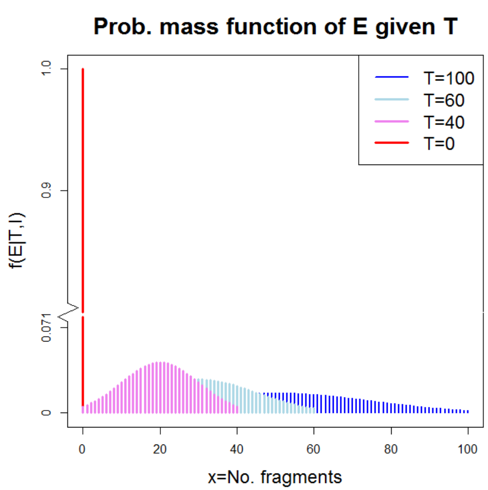
H_1= "Mr G smashed the window"

H_2: "Someone (or something) else smashed the window".



H	Probability
H_1	$P(\mathrm{H} = H_1 \vert I)$
H_2	$P(\mathrm{H} = H_2 \vert I)$





Once the value of T is known the state of H is no longer relevant for the state of E.

Influence diagrams

Decision-theoretic components can be added to a Bayesian network. The complete network is then related to as a *Bayesian Decision Network* or more common ***Influence diagram* (*ID*)**

Method of detection gives a positive result (detection) in 99 % of the cases when the dye is present, i.e. the proportion of false negatives is 1% and a negative result in 98 % of the cases when the dye is absent, i.e. the proportion of false positives is 2%

The presence of dye is rare: prevalence is about 0.1 %

Return to the example with banknotes.

Let

H_0: Dye is present

H_1: Dye is not present

States of the world

and let

E_1: Method gives positive detection

E_2: Method gives negative detection

Data

Assume that…

- The banknote is a SEK 100 banknote
- If we deem the banknote to have been contaminated with the dye, we will consider it as useless and it will be destroyed
- If we deem the banknote not to have been contaminated with the dye, we will use it (in the future) for ordinary purchasing
- Upon using the banknote for purchasing, if it is revealed (by other means than our method) that the banknote is contaminated with the dye, there is a fine of SEK 500

Hence, a payoff function for this problem is

Action	State of the world	
	Dye is present (H_0)	Dye is not present (H_1)
Destroy banknote	0	–100
Use banknote	–500	0

Note that the amounts of money should be entered as negative payoffs. If our utilities are linear in money, this is also our (dis)utility function

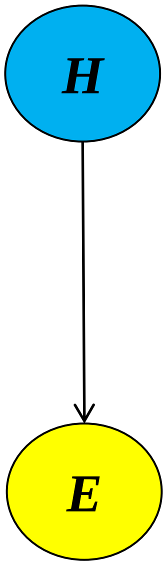
We may however consider a loss function to better describe the situation.

$$\text{Recall: } L(a, \boldsymbol{\theta}) = \max_{a' \in \mathcal{A}} \big(U(a', \boldsymbol{\theta})\big) - U(a, \boldsymbol{\theta})$$

Action	State of the world	
	Dye is present (H_0)	Dye is not present (H_1)
Destroy banknote	0 – 0 = **0**	0 – (–100) = **100**
Use banknote	0 – (–500) = **500**	0 – 0 = **0**

but is this description so much better than the one with (dis)utilities?

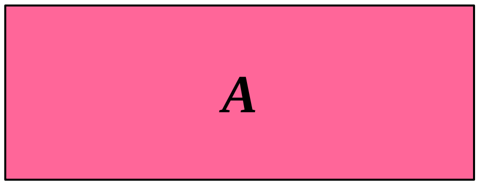
A simple Bayesian network can be constructed for the relevance between the state of the world and data:



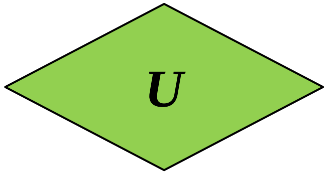
H	*Probabilities*
H_0	0.001
H_1	0.999

		Probabilities	
	H:	H_0	H_1
E	E_1	0.99	0.02
	E_2	0.01	0.98

Now, we will add two nodes to the network, one for the actions that can be taken and one for the utility function



A	
a_1	Destroy banknote
a_2	Use banknote



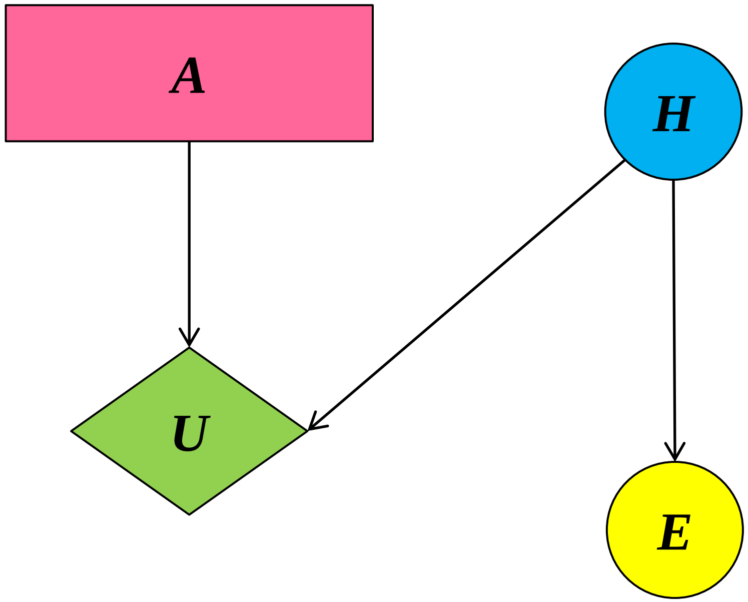
H:	H_0		H_1	
A:	a_1	a_2	a_1	a_2
U	0	–500	–100	0

H_0: Dye is present

H_1: Dye is not present

Neither of the nodes are random nodes.

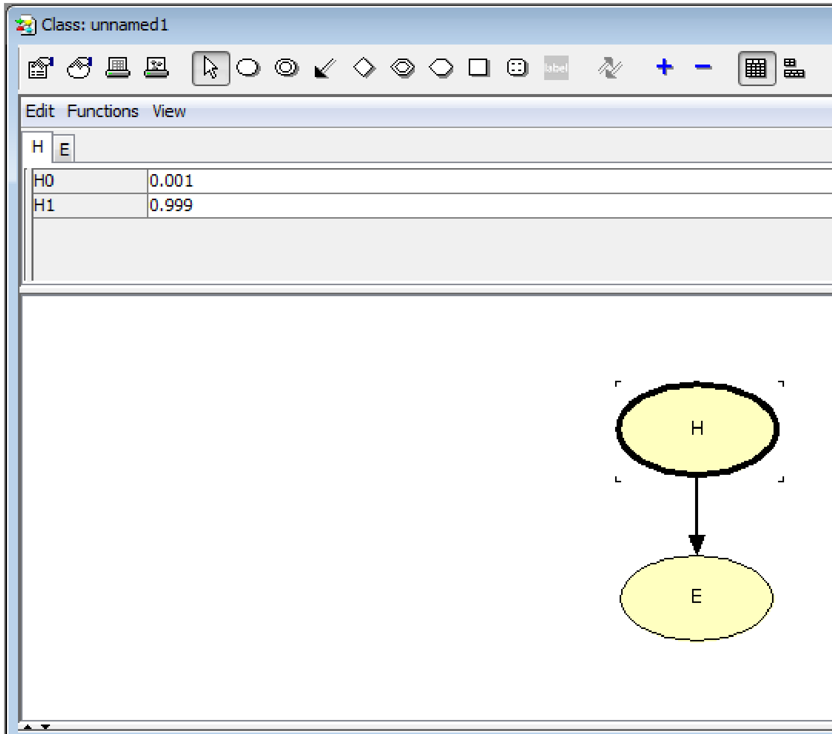
Node ***U*** must be a child node with nodes ***H*** and ***A*** as parents.



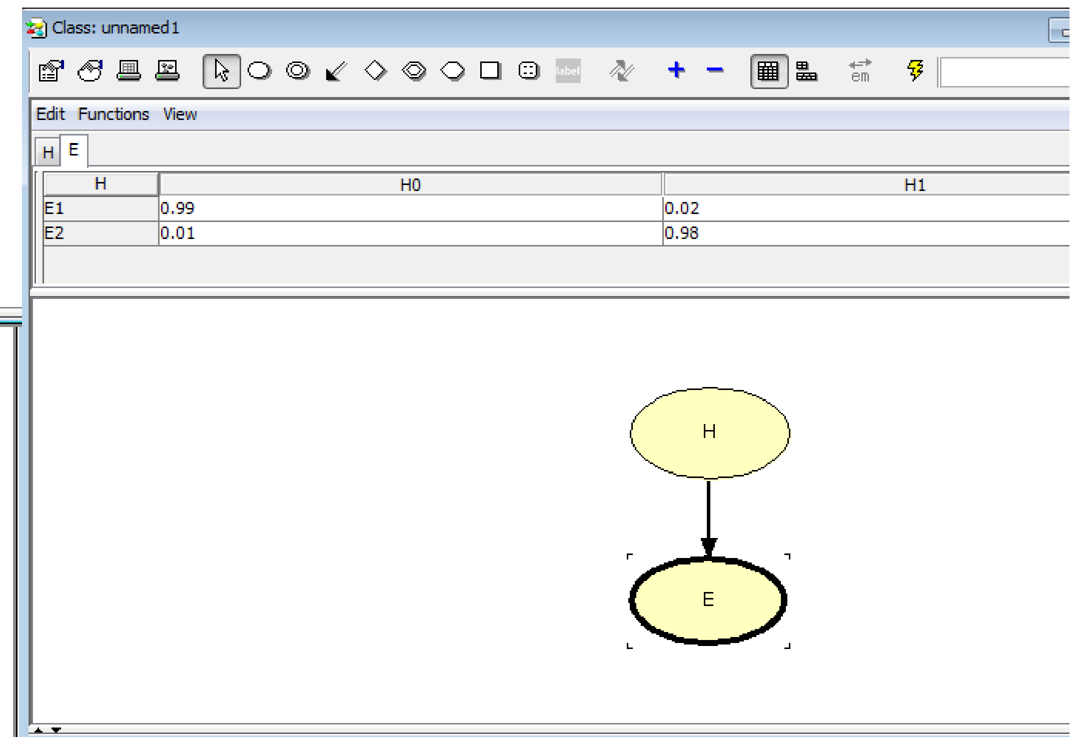
With this network, an influence diagram, we would like to be able to propagate data from node E to a choice of decision in node A.

Hence, in node A the posterior expected utility should be calculated, and the utilities should be specified in node L.

Using Hugin:

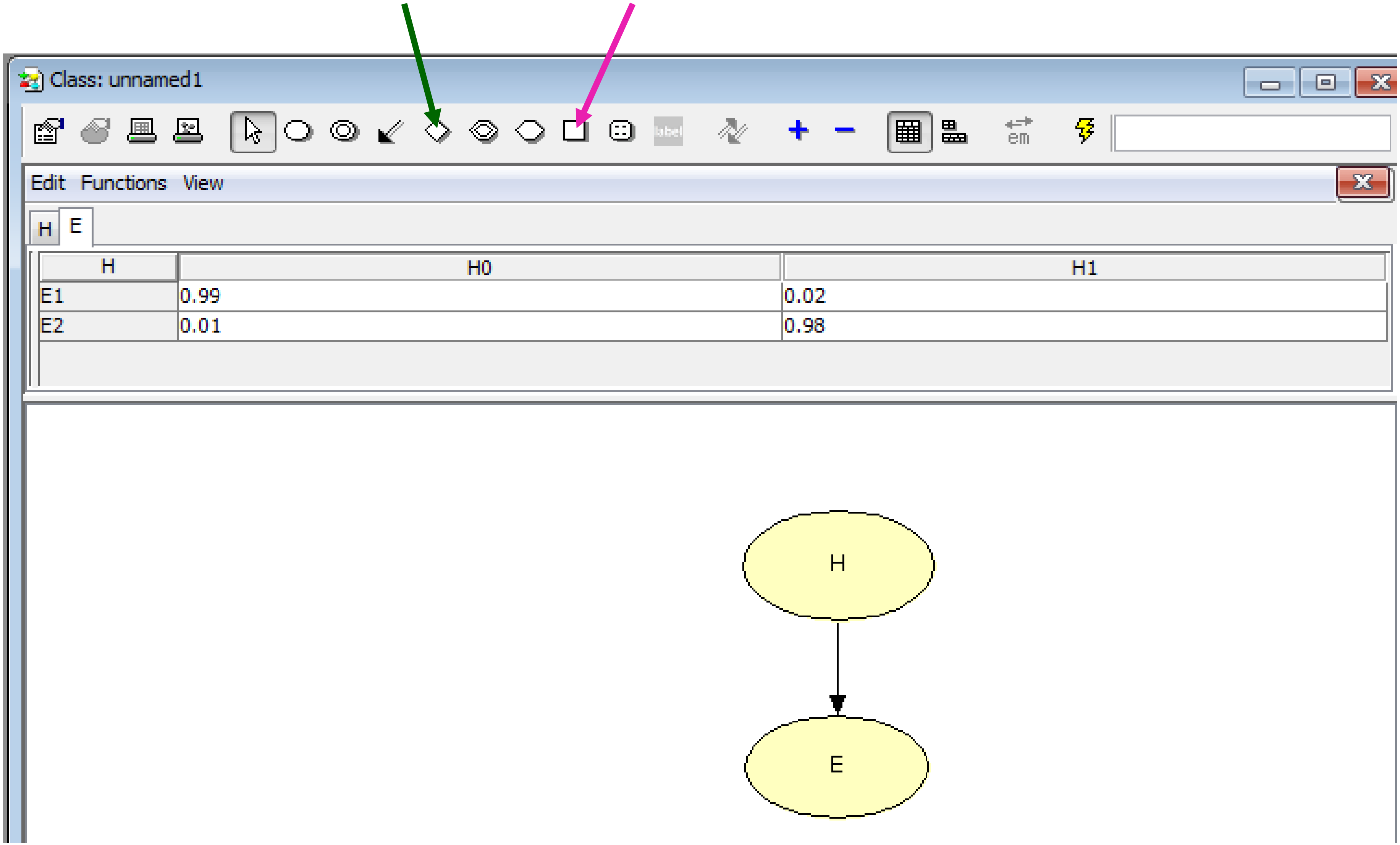


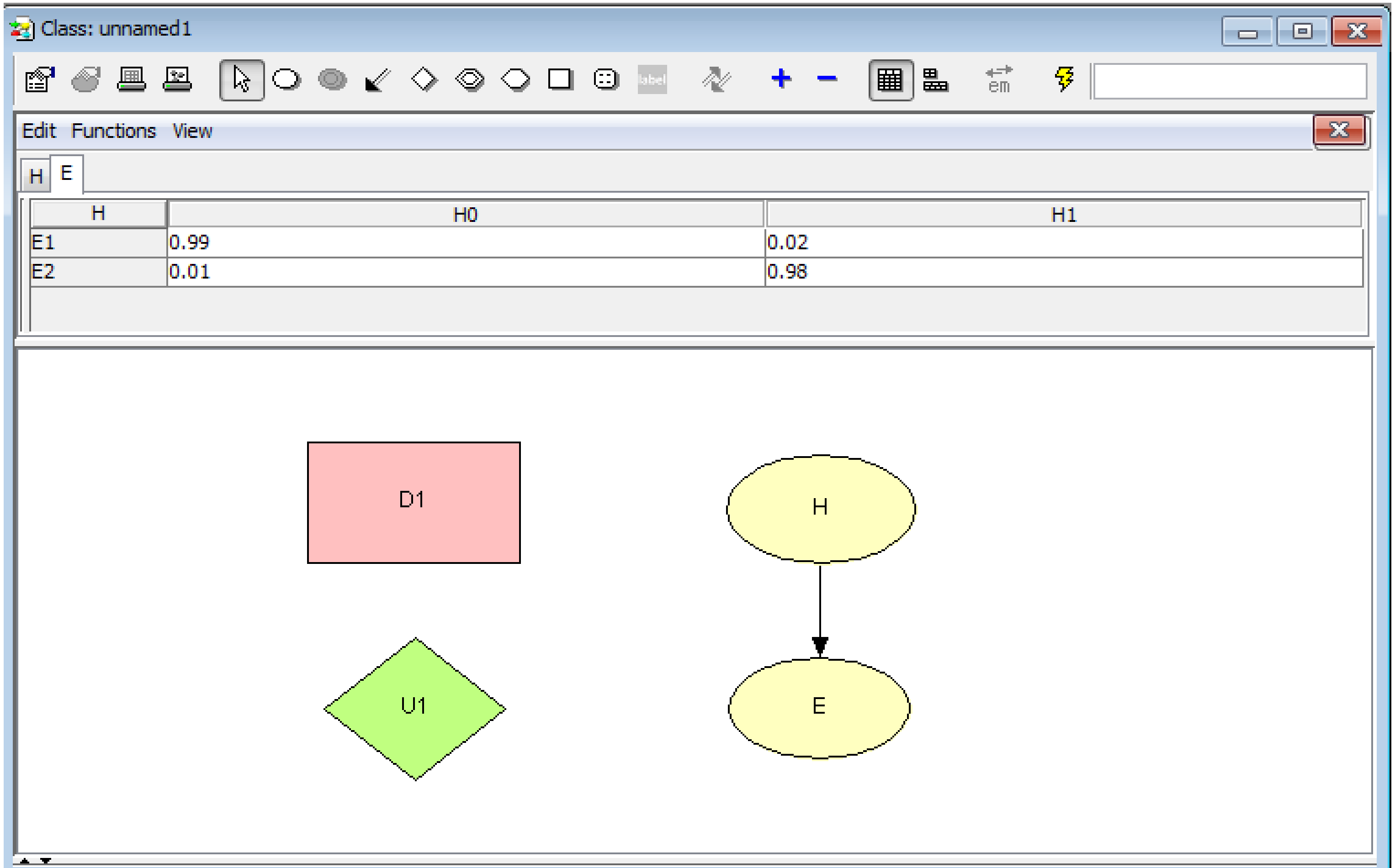
Node *H* with probability table



Node *E* with probability table

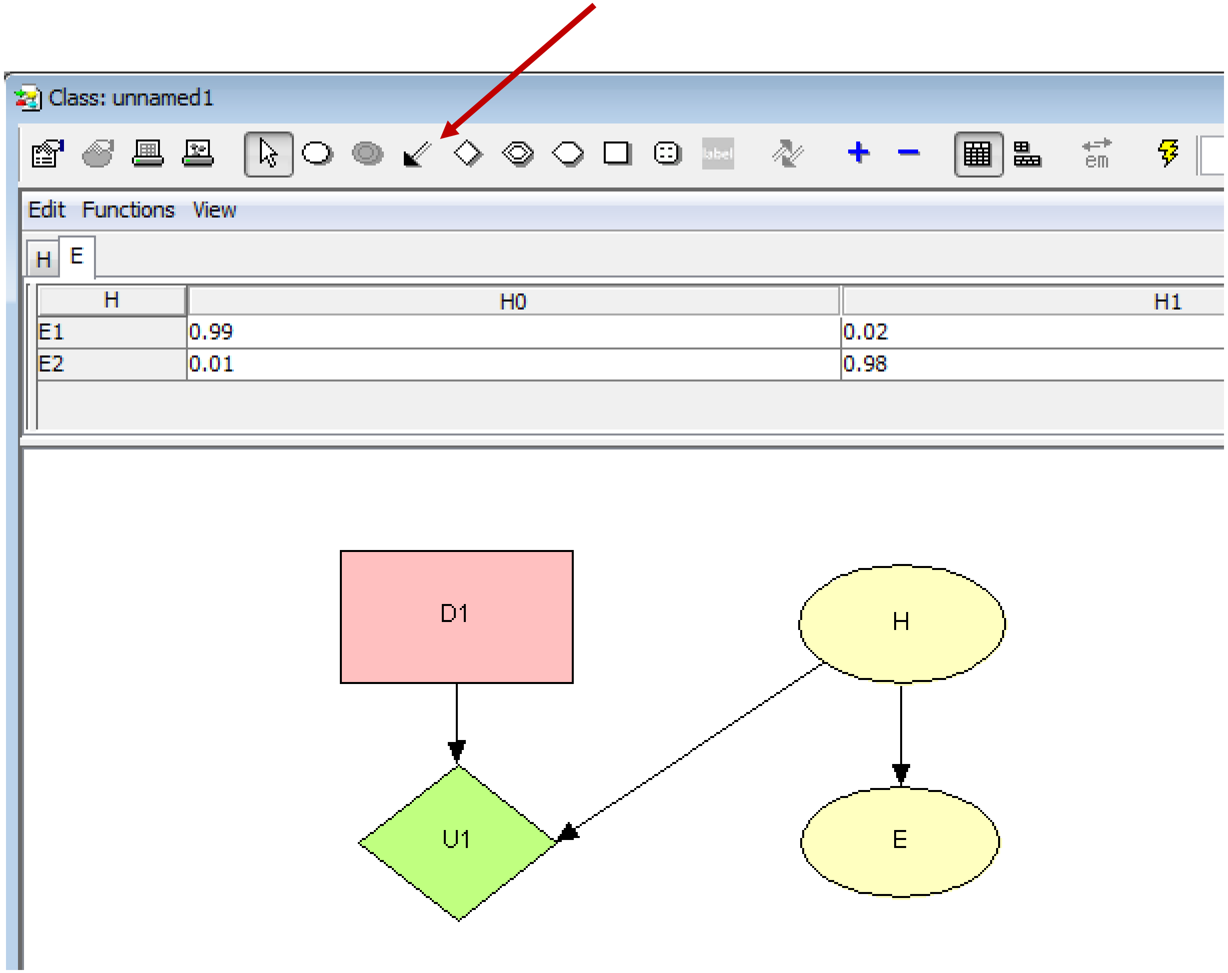
Now, nodes for action and utility should be added.



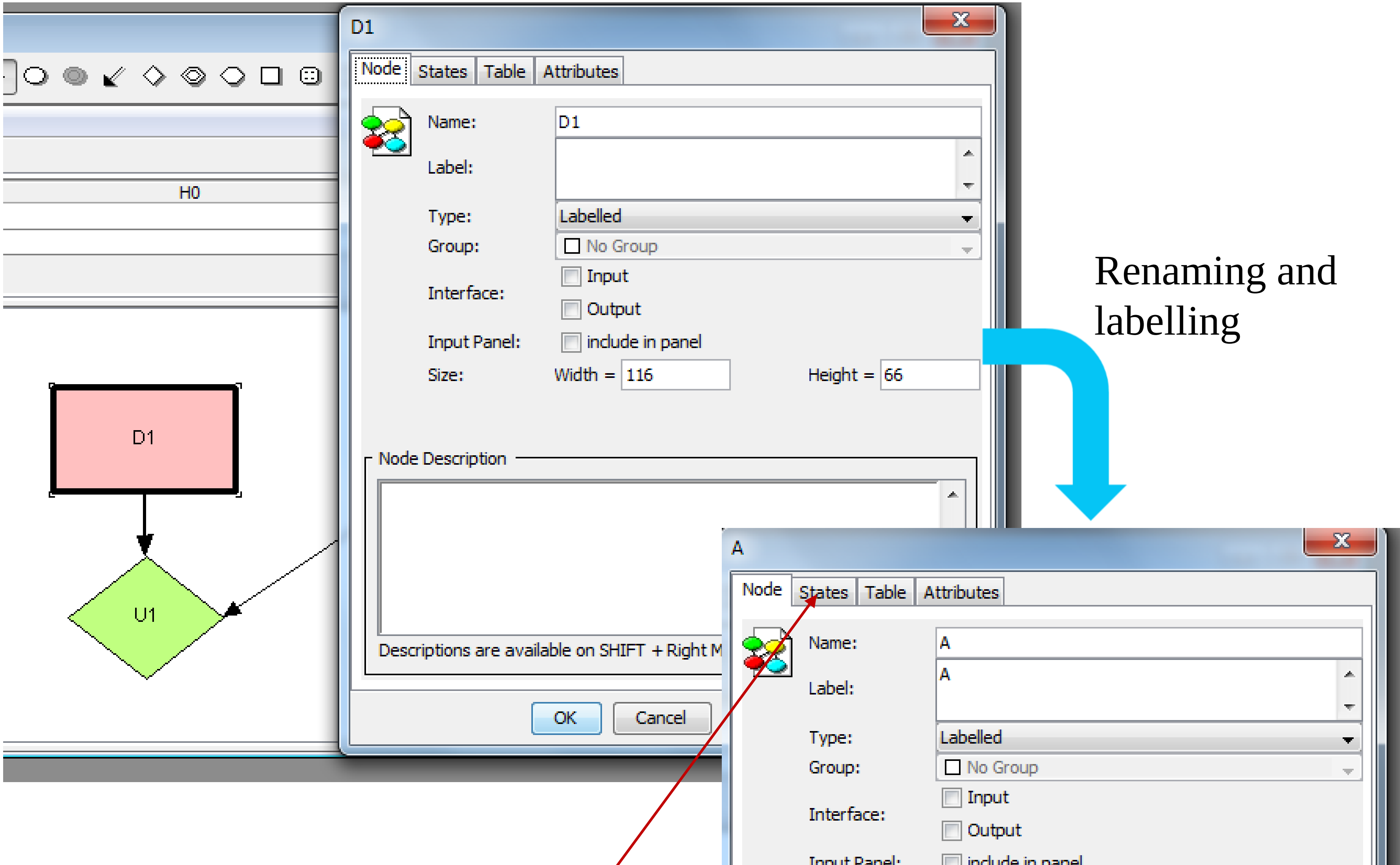


Action nodes by default gets a name with D (for decision) and number.
Utility nodes by default gets a name with U and a number.

Add the links (drag from interior of parent node to interior of child node).



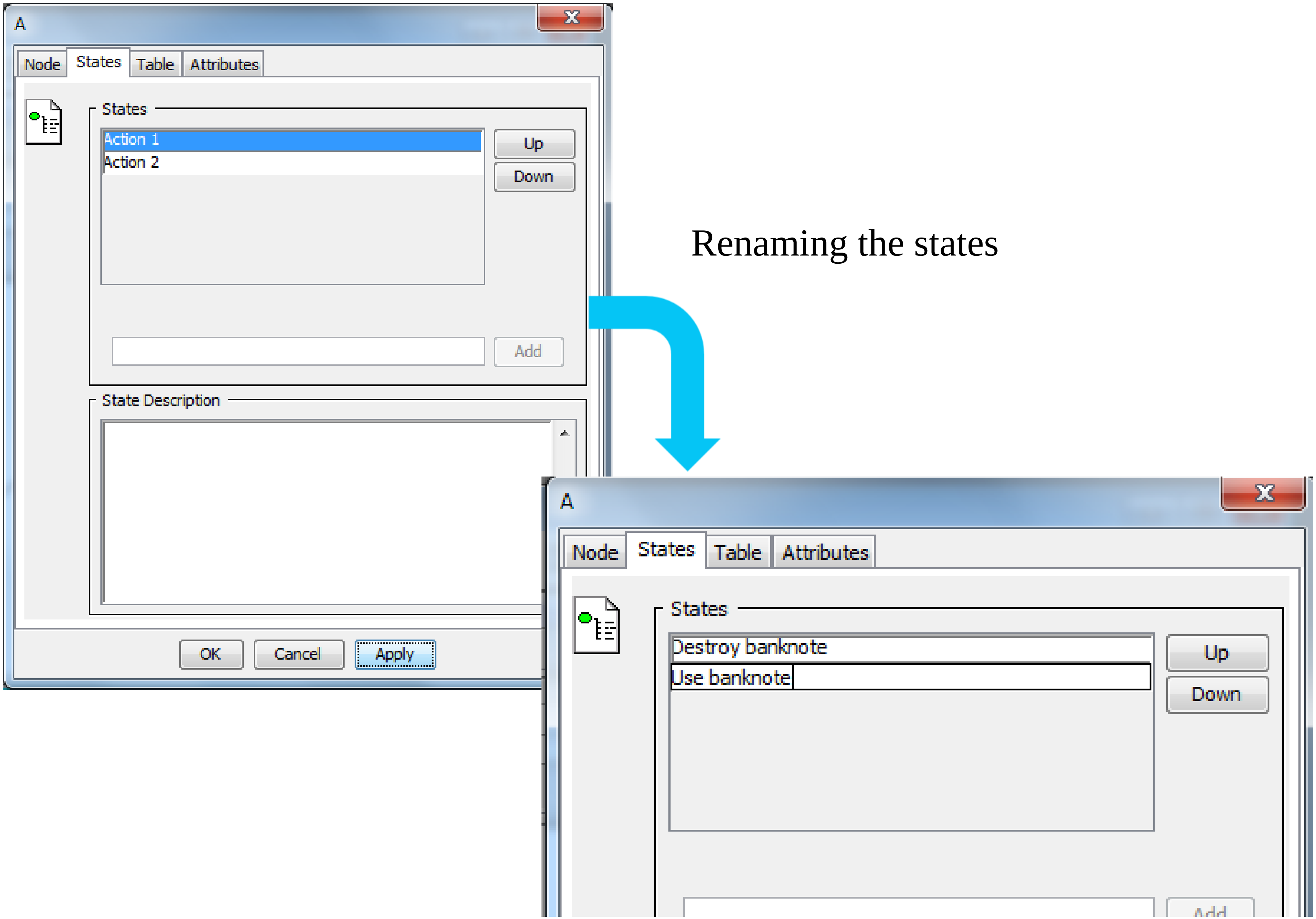
Now, double-click on the action node.



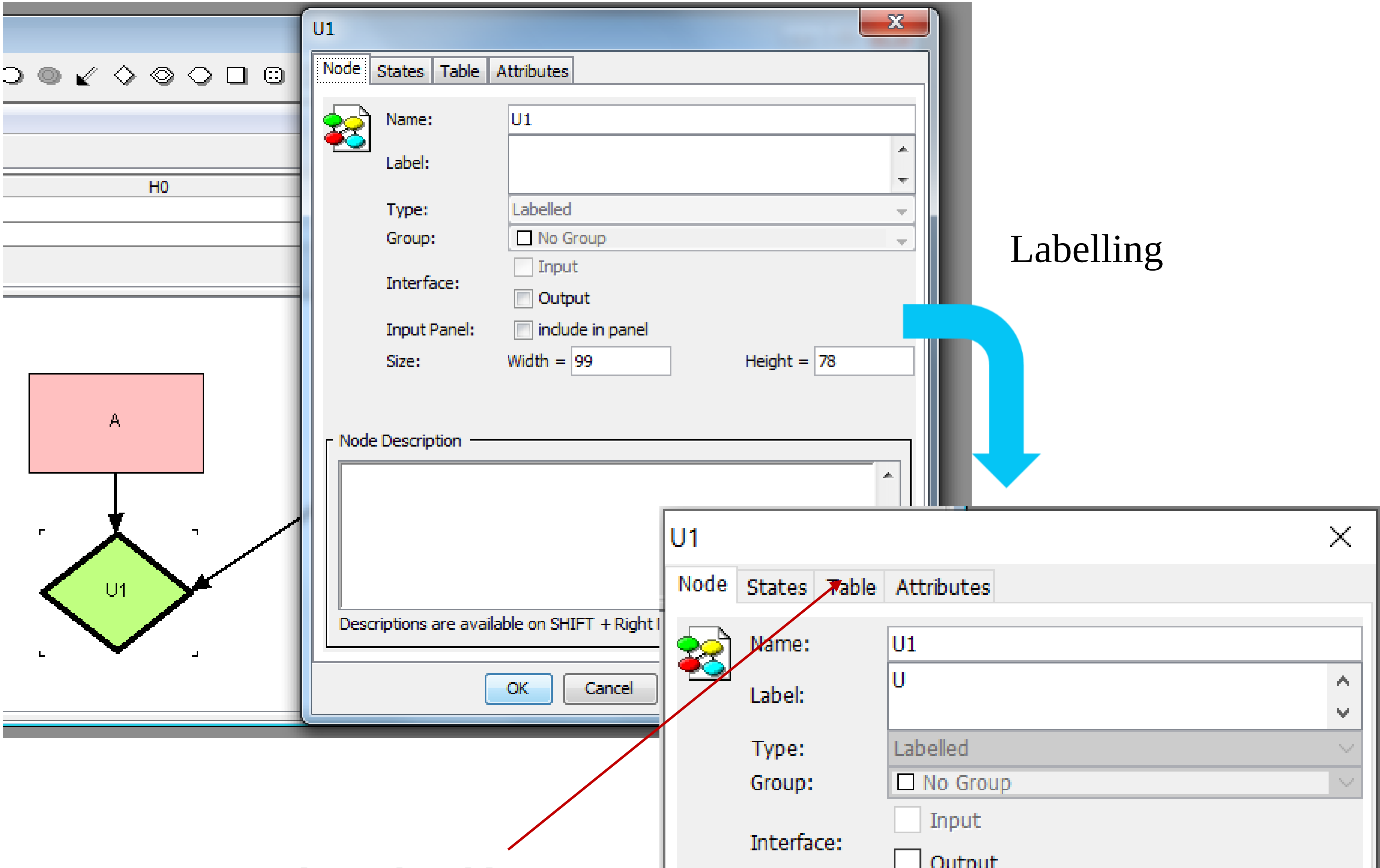
Renaming and labelling

Select tab States.

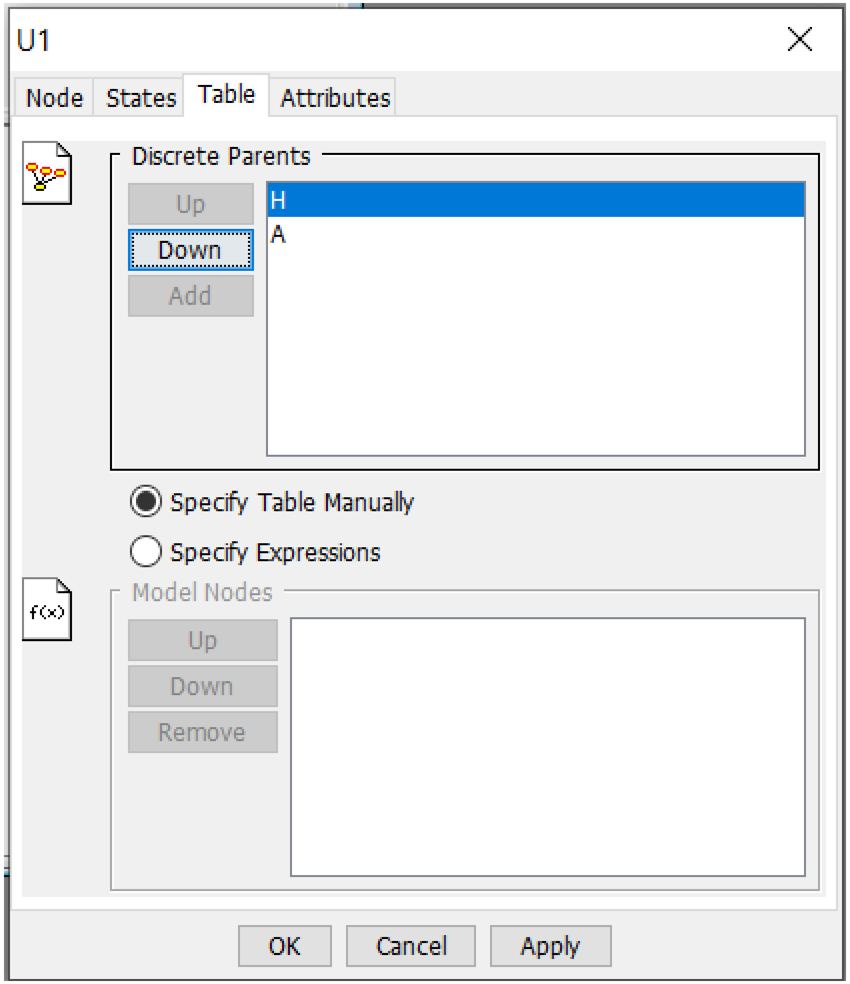
Renaming the states



Double-clicking the utility node…



Select tab Table.



Here, we can set the preference order in which the states of H and states of A should appear in the utility table.

The “ordinary” two-way table…

	H_0	H_1
a_1	$U(a_1, H_0)$	$U(a_1, H_1)$
a_2	$U(a_2, H_0)$	$U(a_2, H_1)$

…will in Hugin appear either as…

H	H_0		H_1	
A	a_1	a_2	a_1	a_2
U (***L***)	$U(a_1, H_0)$	$U(a_2, H_0)$	$U(a_1, H_1)$	$U(a_2, H_1)$

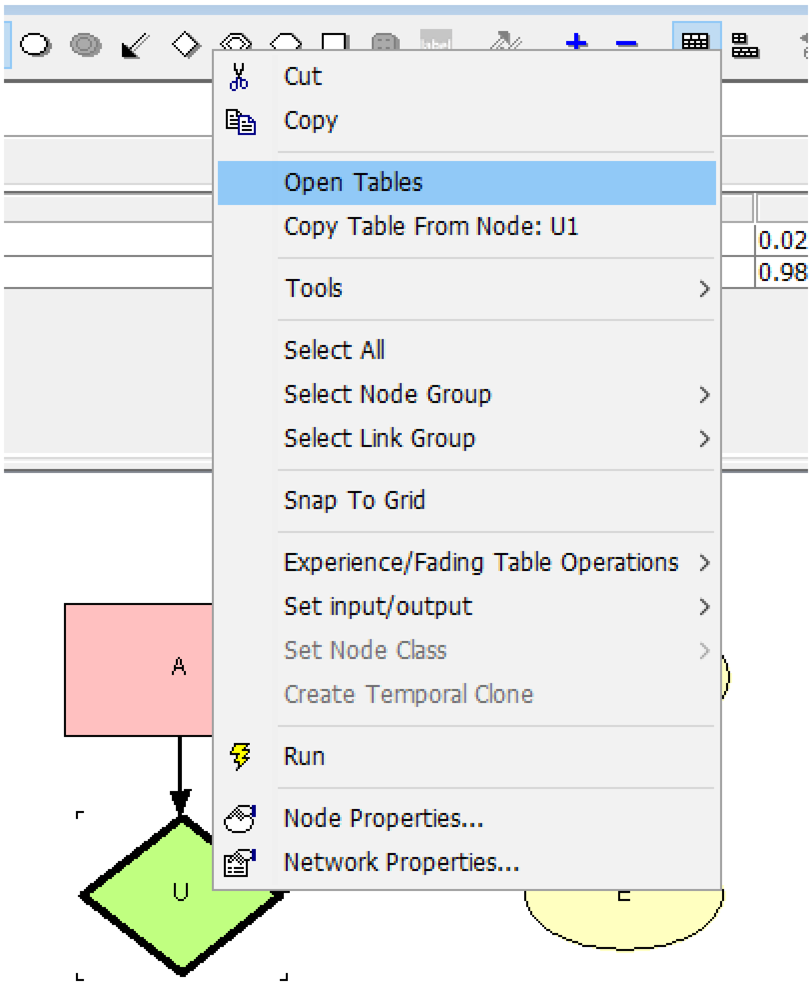
…or as…

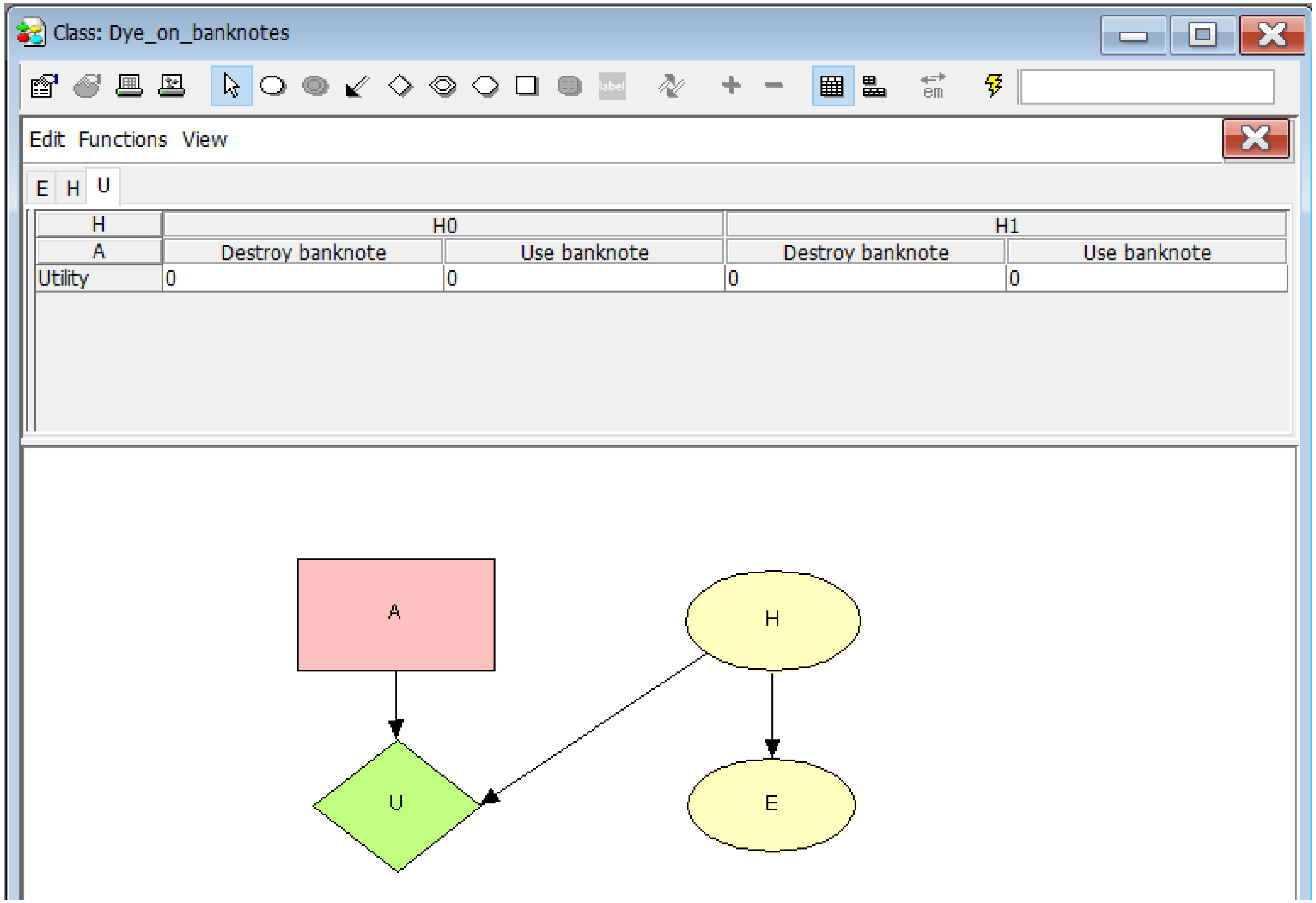
A	a_1		a_2	
H	H_0	H_1	H_0	H_1
U (***L***)	$U(a_1, H_0)$	$U(a_1, H_1)$	$U(a_2, H_0)$	$U(a_2, H_1)$

…depending on the order set.

Here, we keep the order as given.

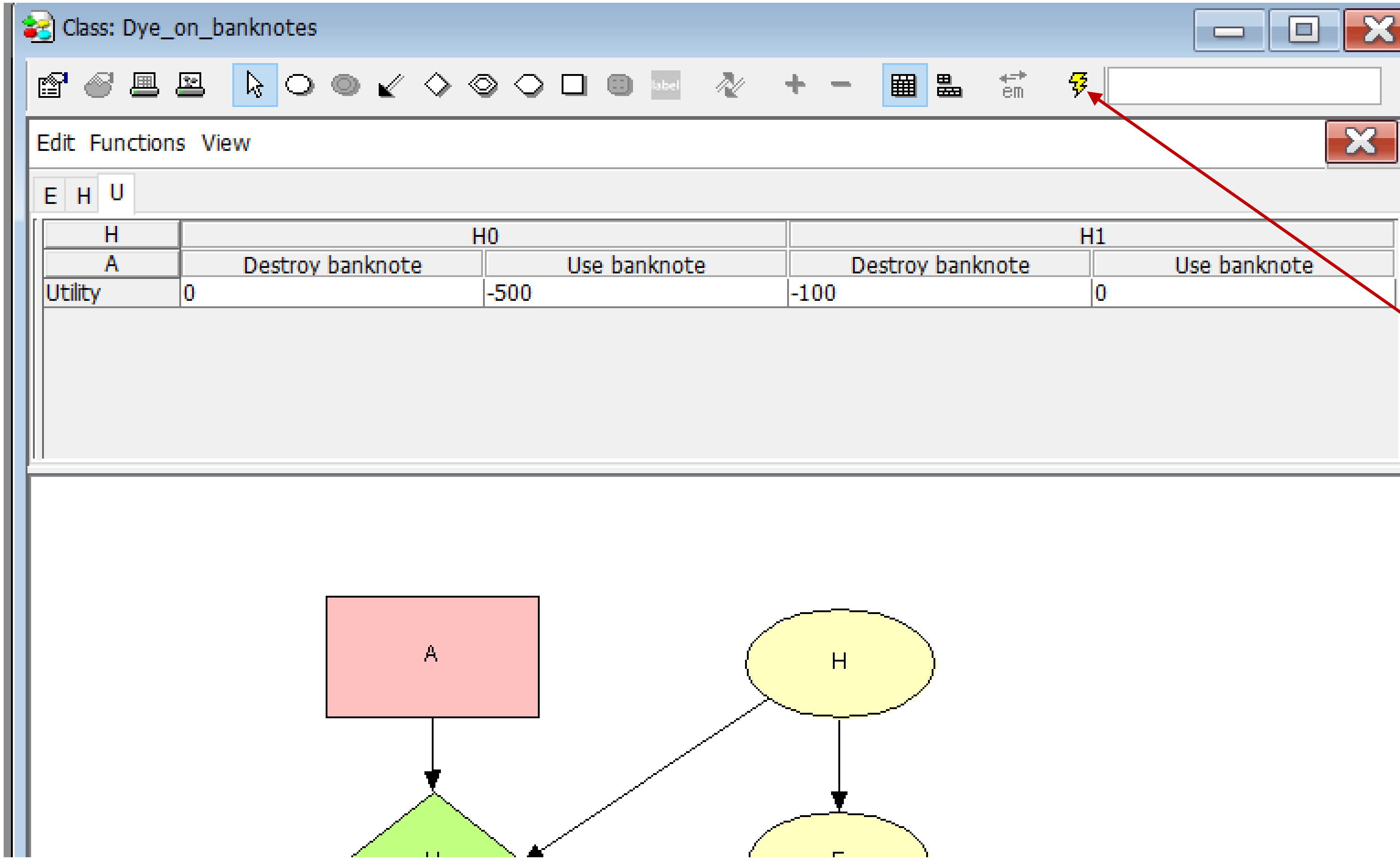
Righ-click on the utility node (U) and select Open Tables.



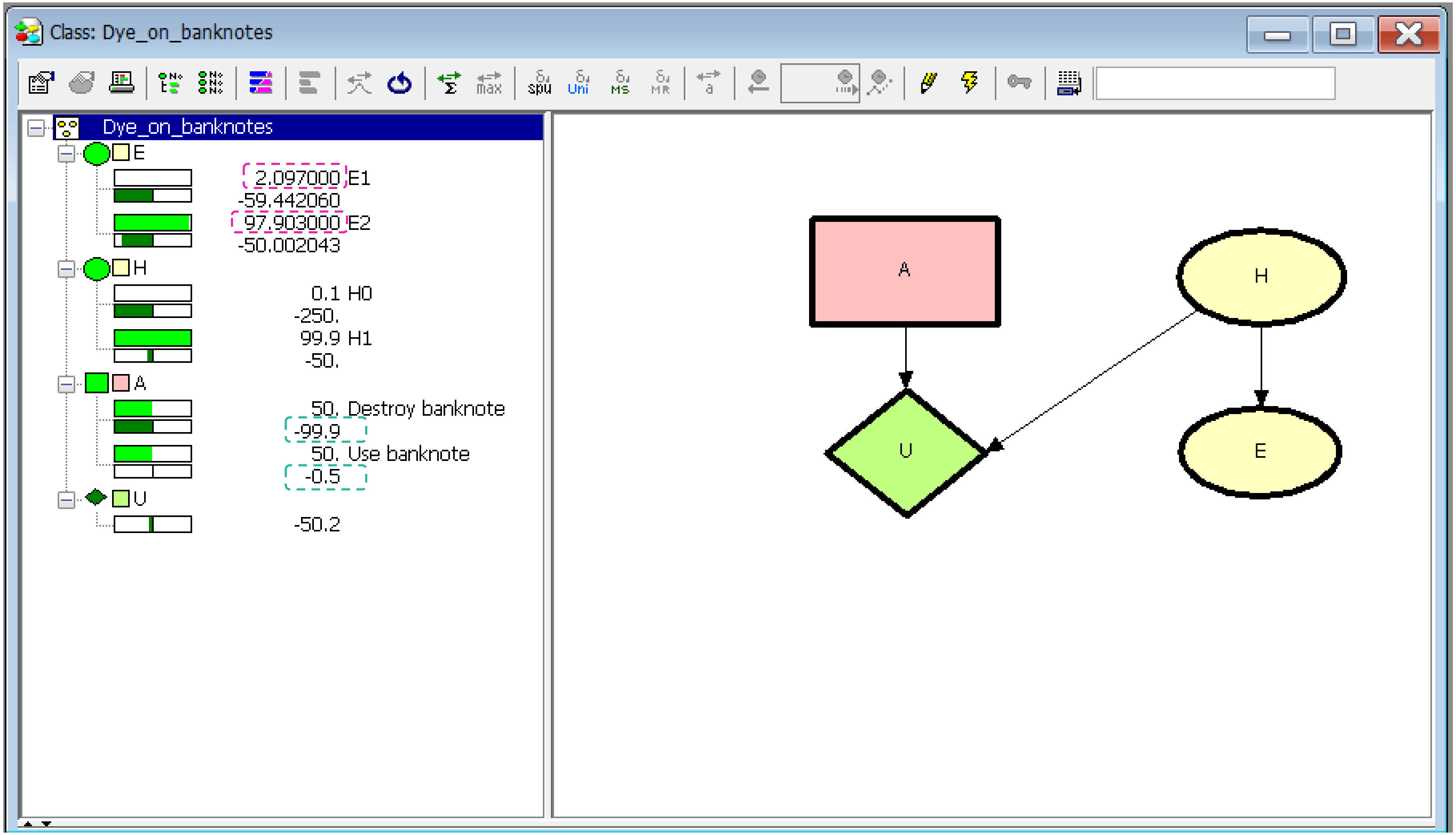


Now, enter manually the (dis)utilities in the table.

H:	H_0		H_1	
A:	a_1	a_2	a_1	a_2
$\boldsymbol{U}$	0	-500	-100	0

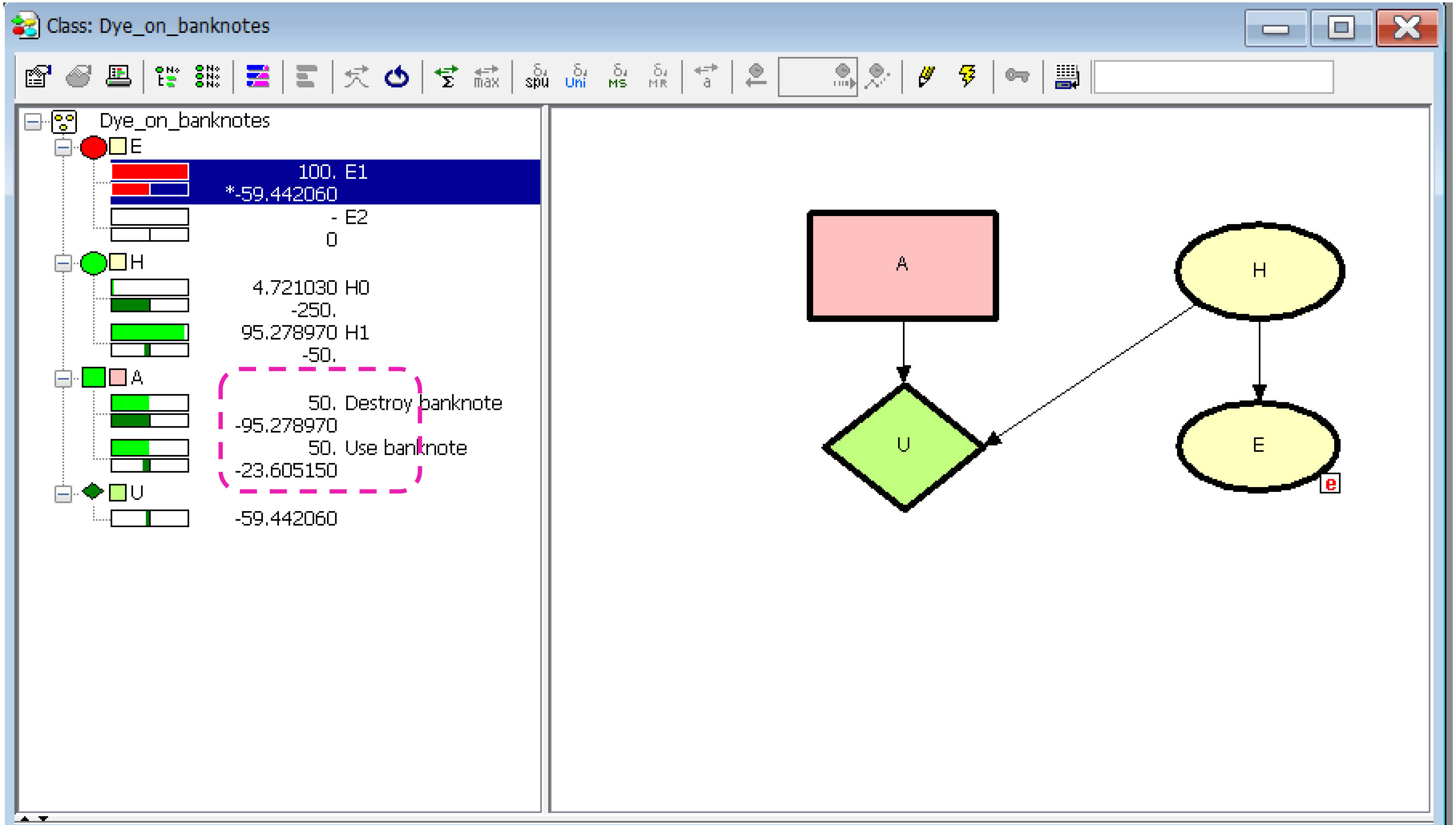


Run the network by clicking the flash icon.



Marginal probabilities and average utilities are displayed.

Now, enter the evidence = “Method gives positive detection” (E_1), by double-clicking on E1.



Here we can read off the calculated expected utilities for each of the two actions. Since the expected (dis)utility of a_1 (Destroy banknote) –95.28 is lower than the expected (dis)utility of a_2 (Use banknote) –23.61, the optimal action is a_2.