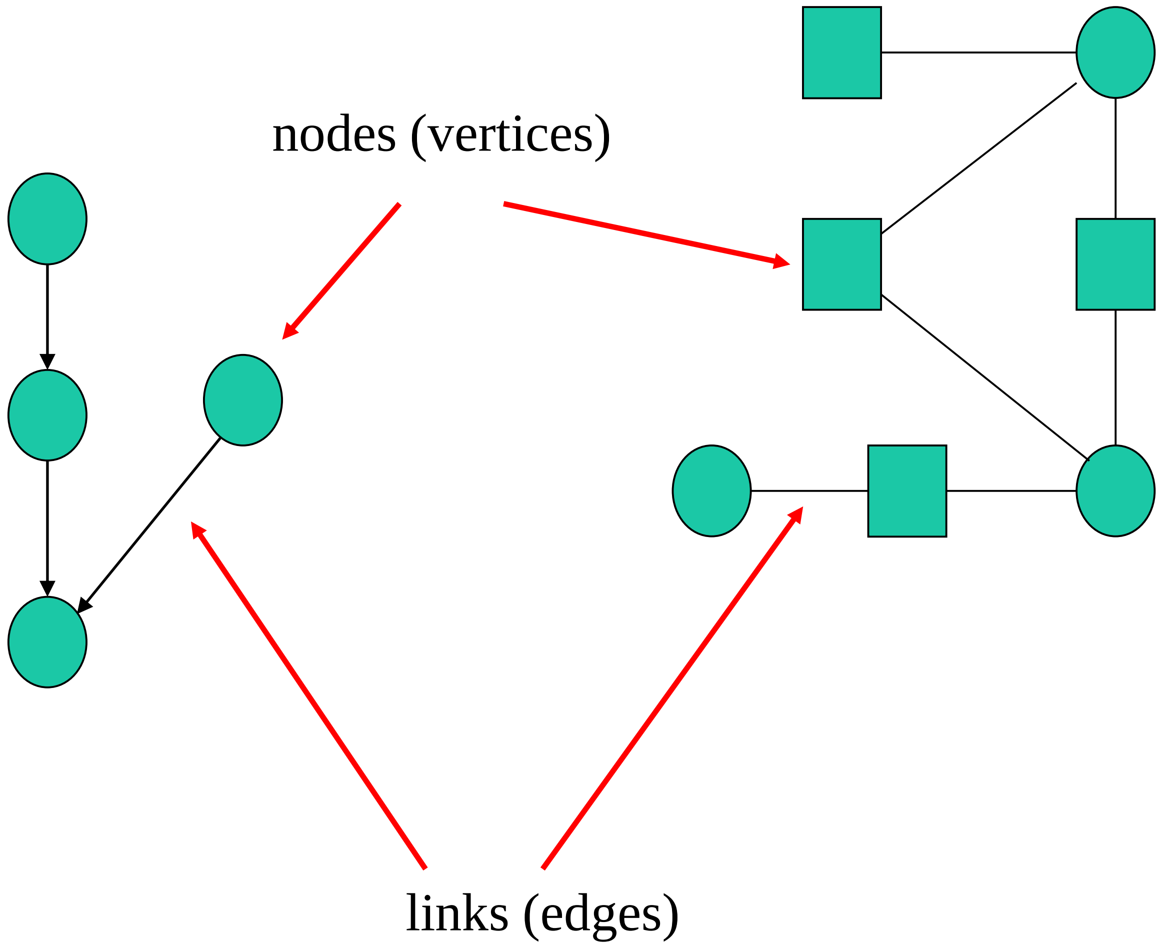
# Meeting 11:
# Graphical models

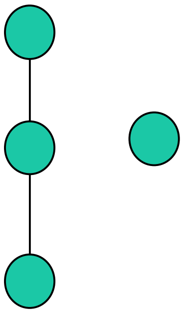
# Bayesian networks

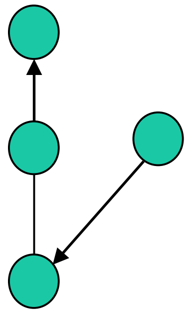
Some general graphical model concepts:

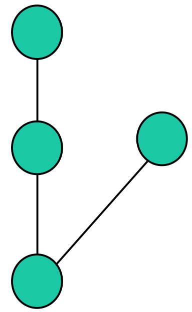


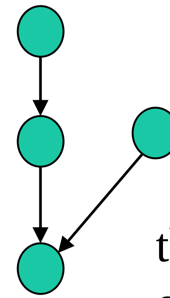
# A graph can be

*disconnected:* or *connected:* ; *undirected:* or *directed:*



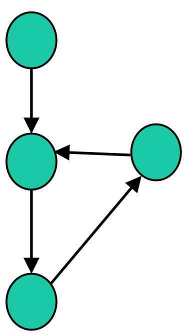




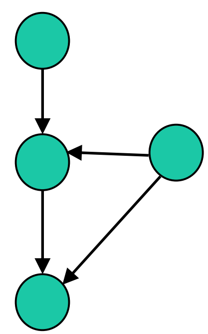


the edges are one-directed arrows

*cyclic:* or *acyclic:*

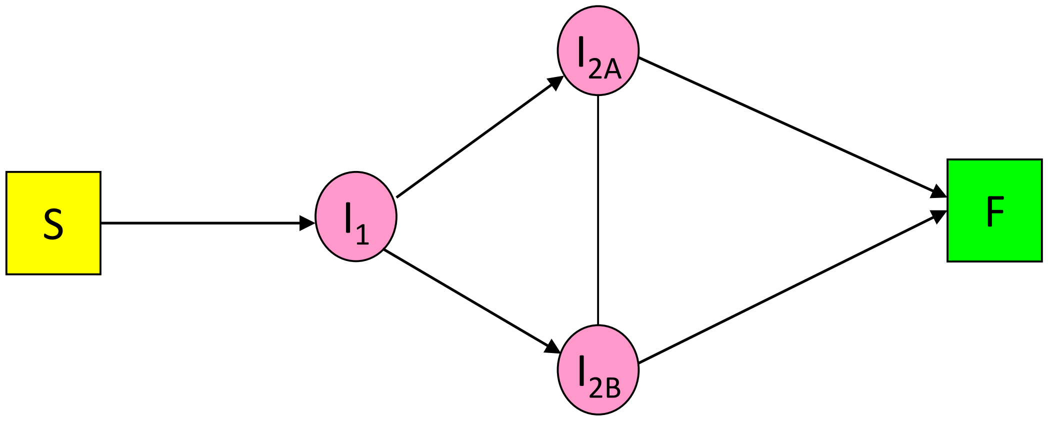


possible to start in one node and “come back”



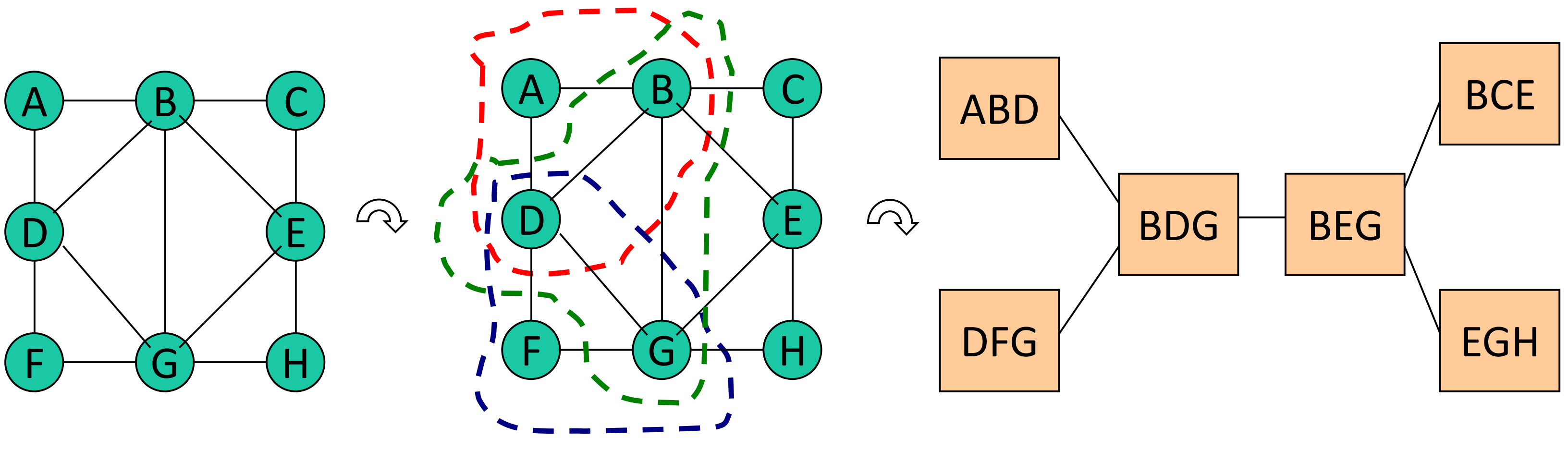
*Examples:*

Transport routes:




Acyclic, but not completely directed

Junction trees:




From 8 nodes to 6 nodes (Source: *Wikipedia)*

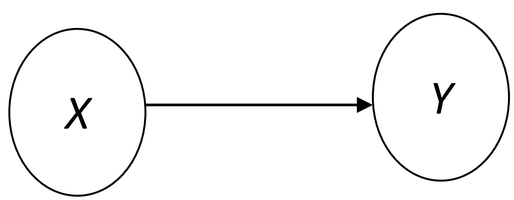
# Bayesian (belief) networks

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

• the nodes (vertices) represent *random variables*

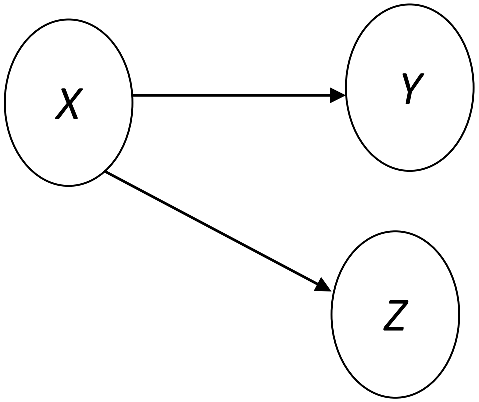
• the links (edges, arcs) represent direct *relevance* relationships among variables

*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

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

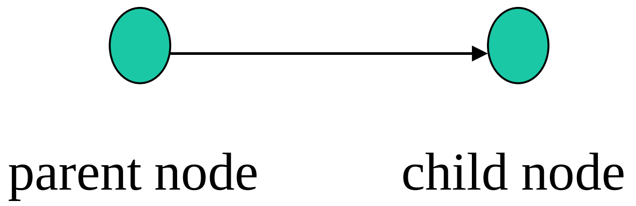
$$\Pr(Z = z \mid X = x, I) \neq \Pr(Z = z \mid I)$$

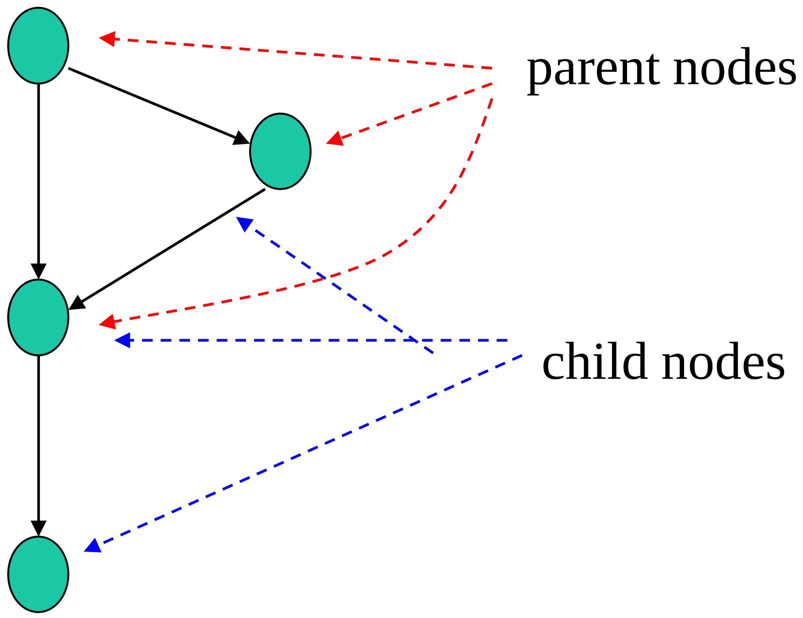
but also (as will be seen below)

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

# Structures in a Bayesian network

There are two classifications for nodes: *parent nodes* and *child nodes*




Thus, a node can be solely a parent node, solely a child node *or* both!

*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.

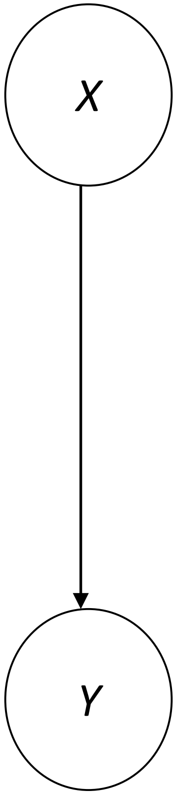
For a node that is *solely* a parent node:

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

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 $x_2$

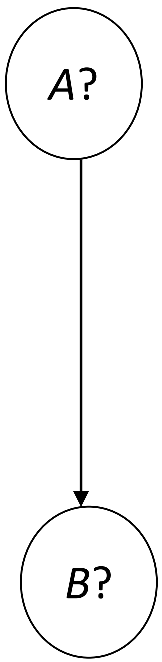
$Y$ has the states $y_1$ and $y_2$

*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"

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

| $A?$ | *Probabilities* |
|---|---|
| $A$ | 0.001 |
| $\overline{A}$ | 0.999 |

Two states:

$B$: "Result is positive"

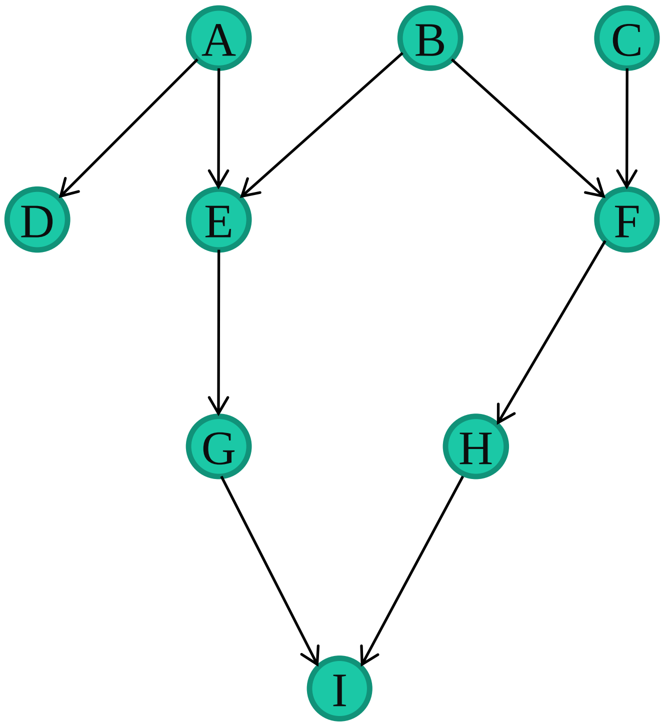
$\overline{B}$: "Result is negative"

| | | *Probabilities* | |
|---|---|---|---|
| $A?$: | | $A$ | $\overline{A}$ |
| $B?$: | $B$ | 0.99 | 0.02 |
| | $\overline{B}$ | 0.01 | 0.98 |

*More about the structure…*

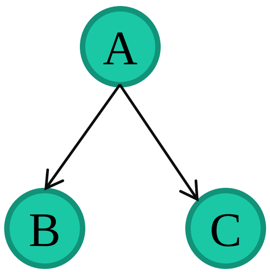
## Ancestors and descendants:

A node $X$ is an *ancestor* of a node $Y$ and $Y$ is in turn a *descendant* of $X$ <u>if</u> there is a <u>unidirectional</u> path from $X$ to $Y$

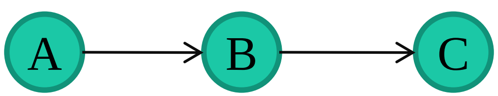



| *Ancestor* | *Descendants* |
|---|---|
| A | D, E, G, I |
| B | E, F, G, H, I |
| C | F, H, I |
| E | G, I |
| F | H, I |
| G | I |
| H | I |

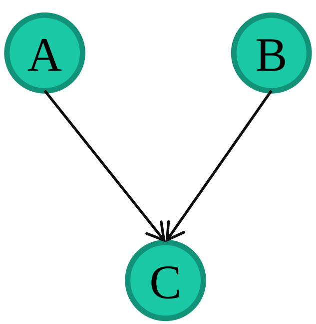
Different connections:




*diverging connection*



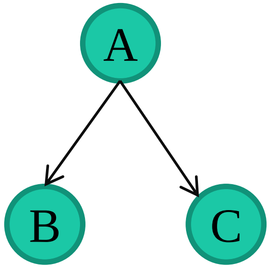

*serial connection*




*converging connection*

# Conditional independence and *d*-separation

1) Diverging connection




There is a path between B and C even if it not unidirectional

➔ B may be relevant for C (and vice versa)

However, if the state of A is known this relevance is lost.: The path is *blocked*

➔ B and C are *conditionally independent* given A

*Example*

Let A be a random node with states

$A_1$ = “Willie is a cat”

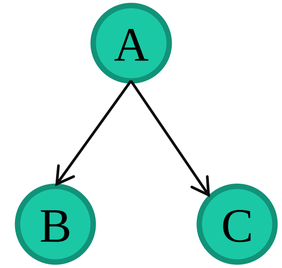
$A_2$ = “Willie is a parrot

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 nib”
$C_2$ = “Willie has no nib”

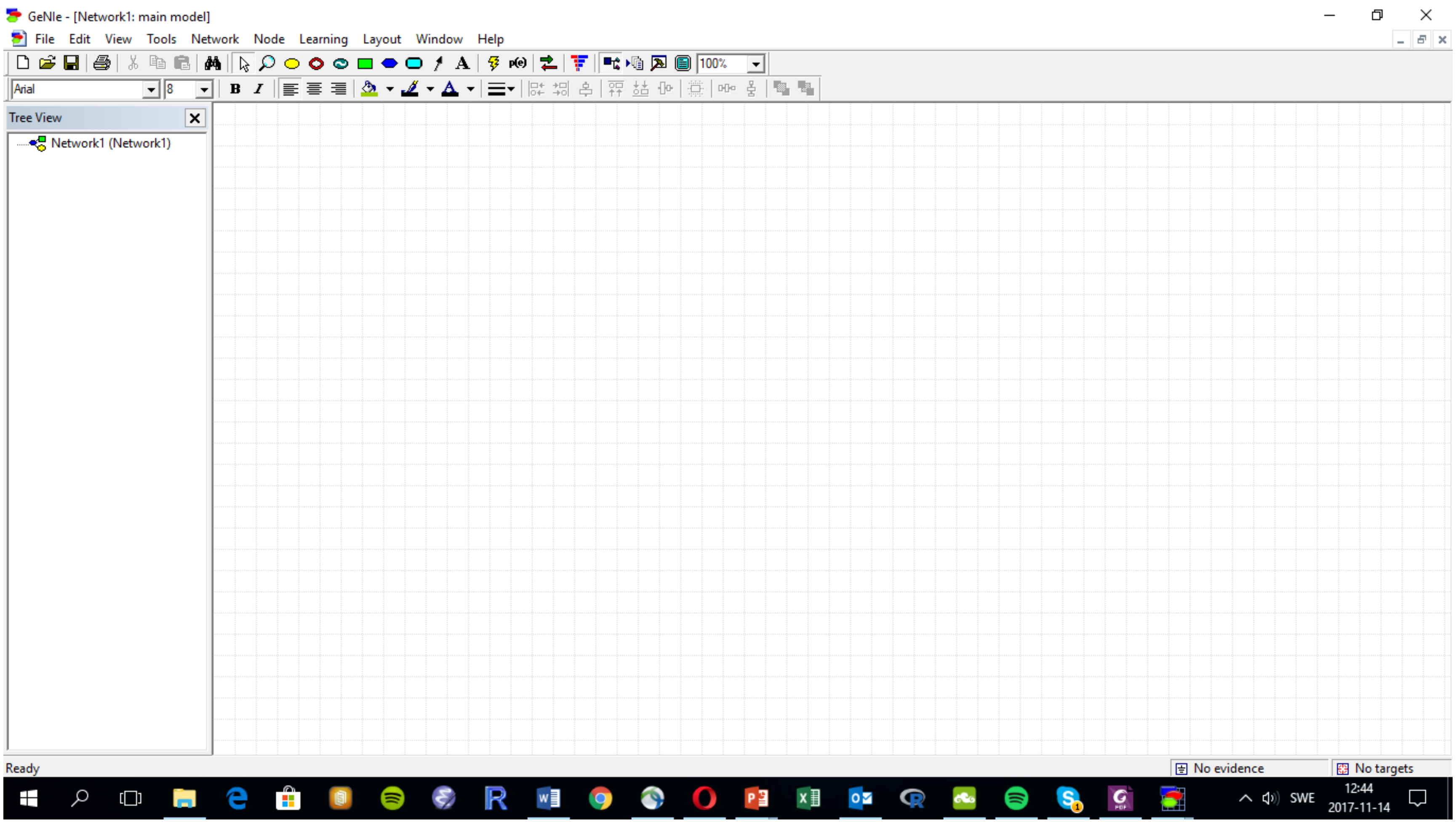



Given B being equal to $B_1$ the conditional probability of $C_1$ is different (lower) than the conditional probability of $C_1$ given B is equal to $B_2$.

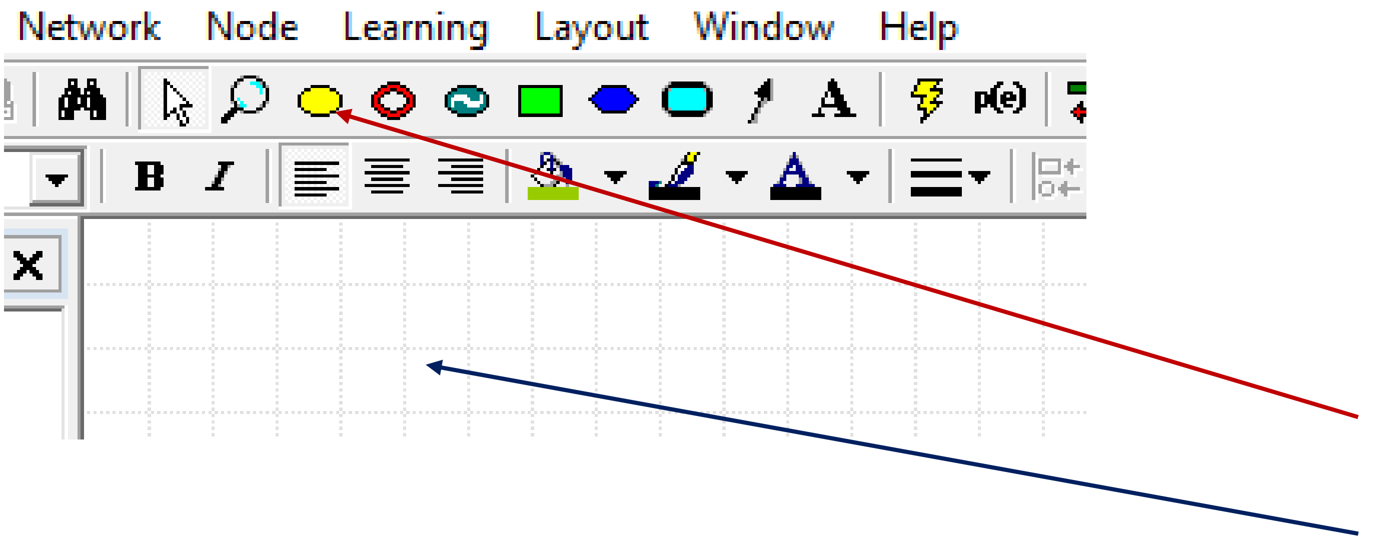
Hence, B is relevant for C and vice versa.

However, if A is instantiated to $A_1$, i.e. Willie is a cat, B and C are no longer relevant for each other if we reasonably assume that the number of legs Willie has cannot affect whether he has a nib or not.

# GeNIe software



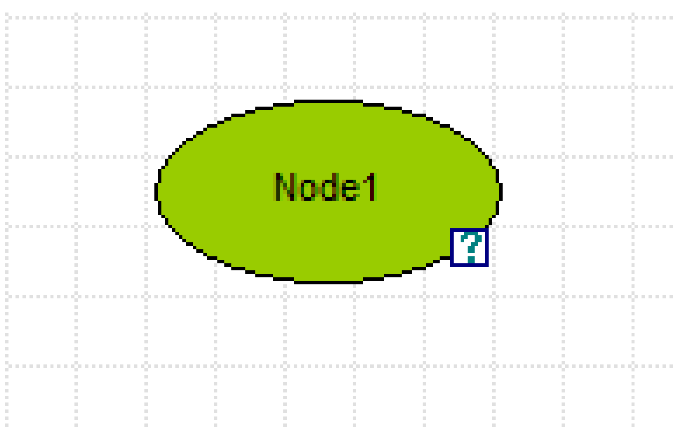
# Adding a chance node

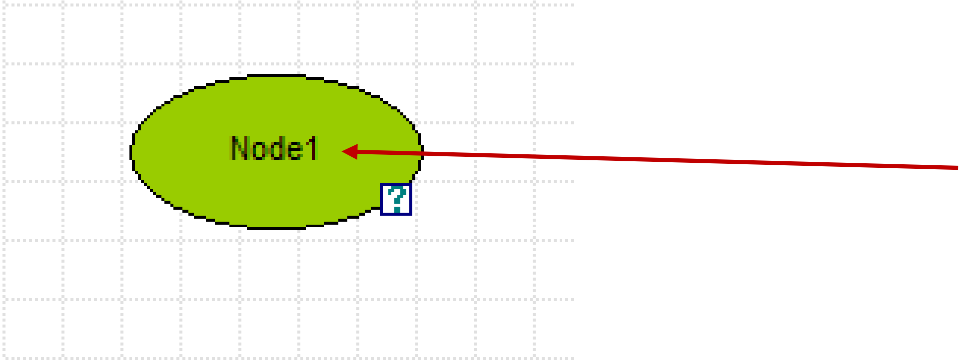



Click here

*then*

Click here

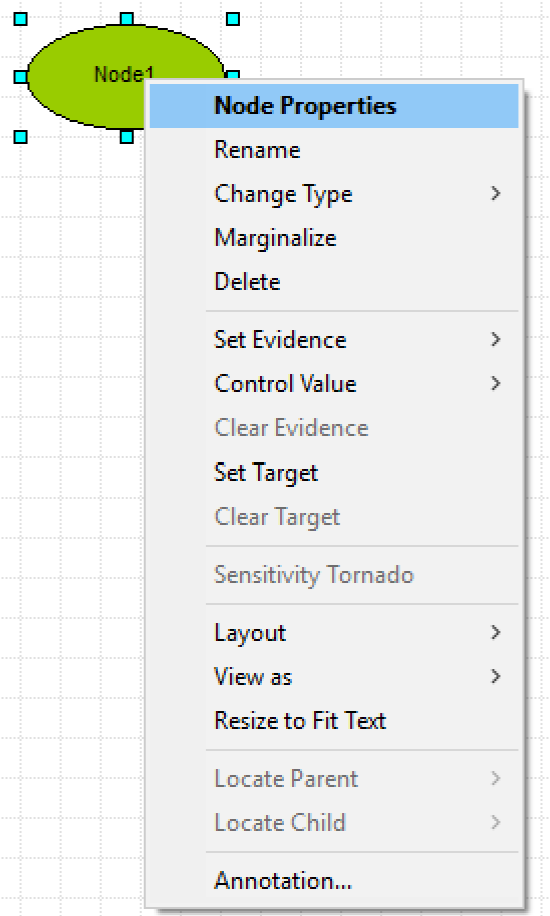




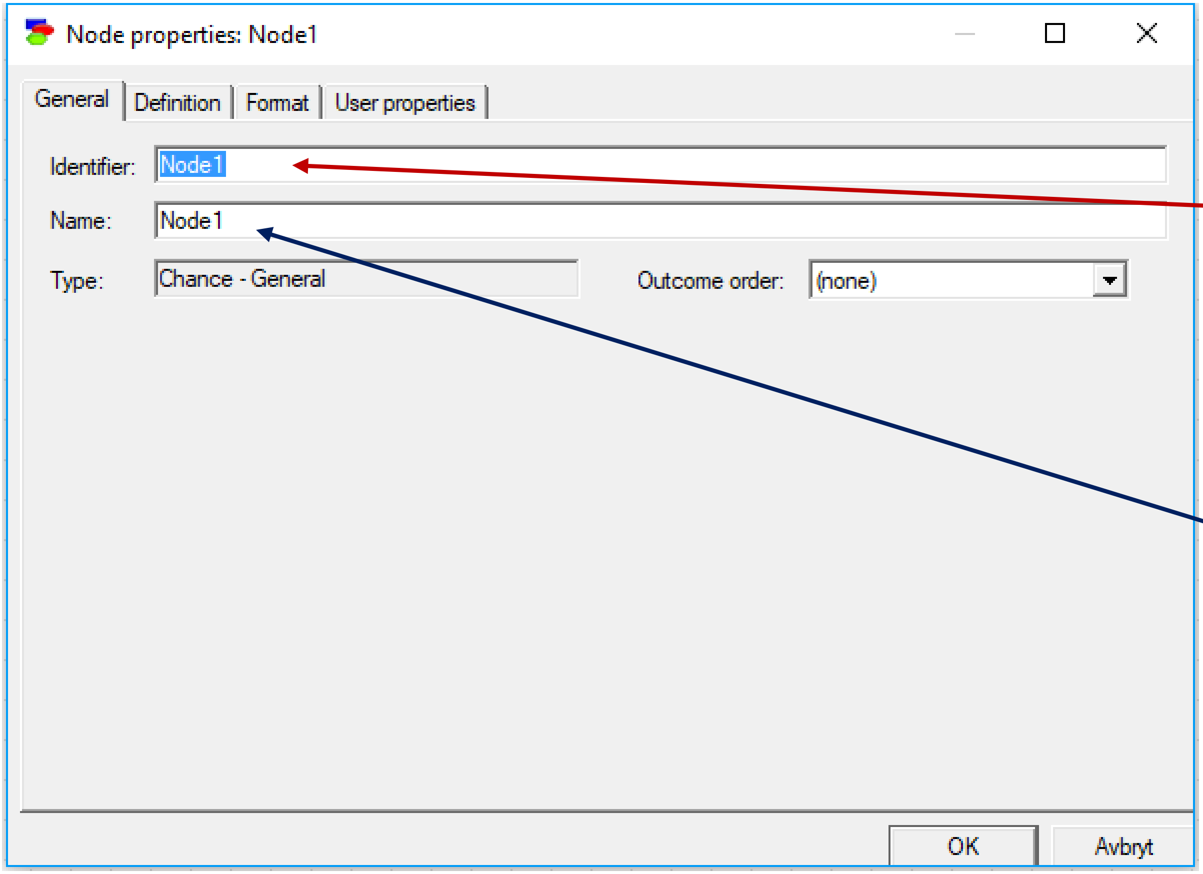

Double-click

*or*

Right-click…




and select
**Node Properties**

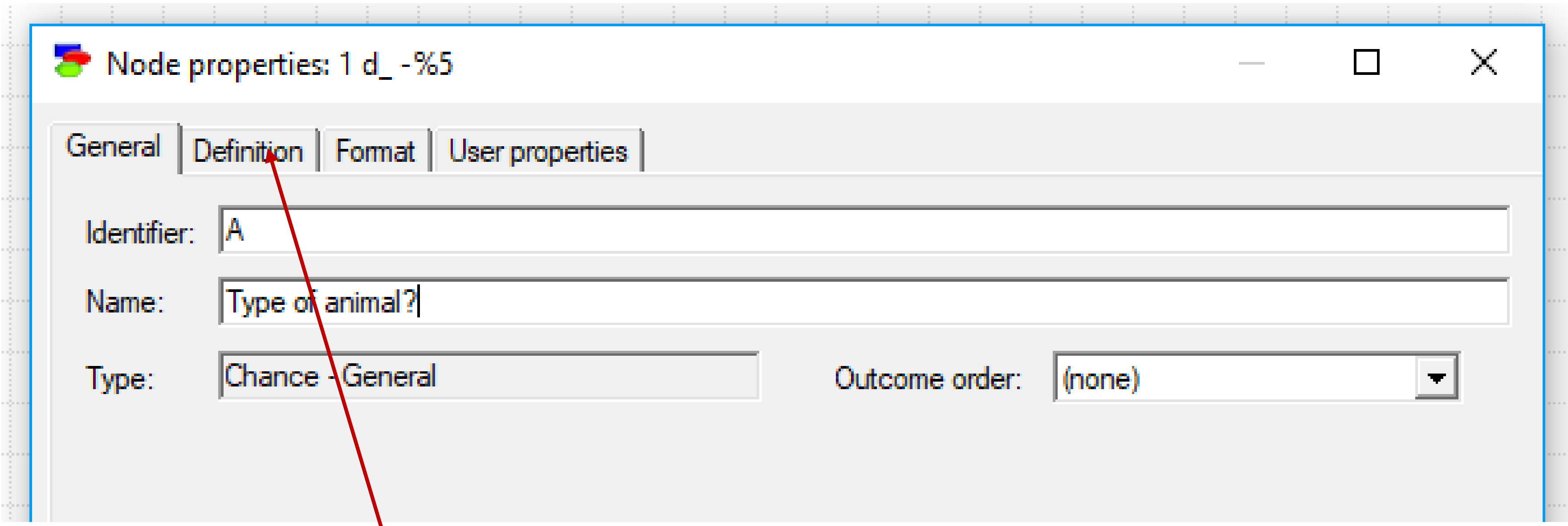



Enter a unique identifier
(One single word starting with a letter and otherwise comprised by letters digits and underscores only)

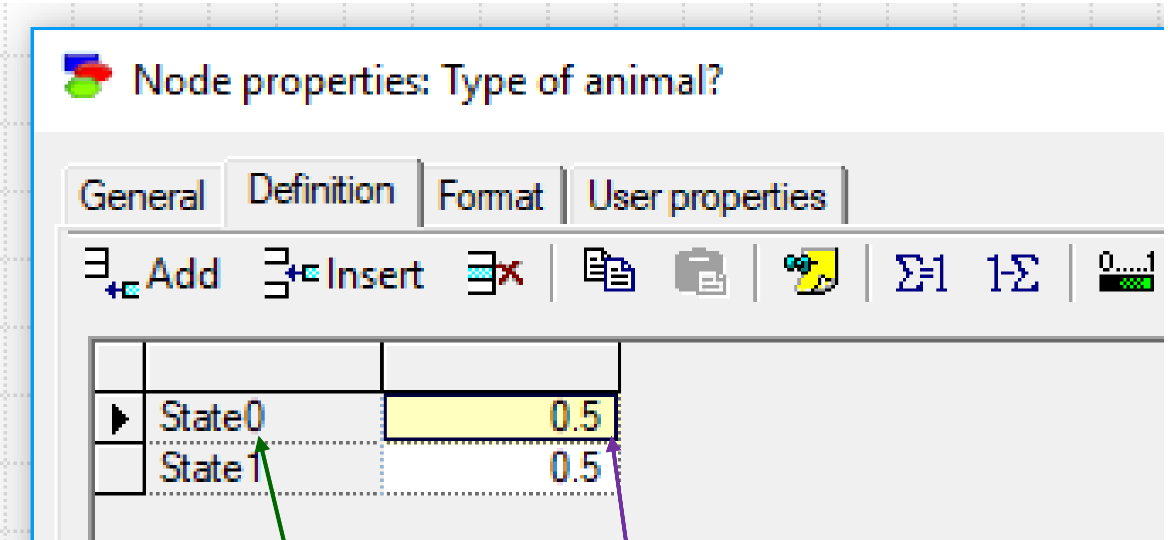
Enter a name (label)
(Free format)

…for instance…

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




Select tab "Definition"

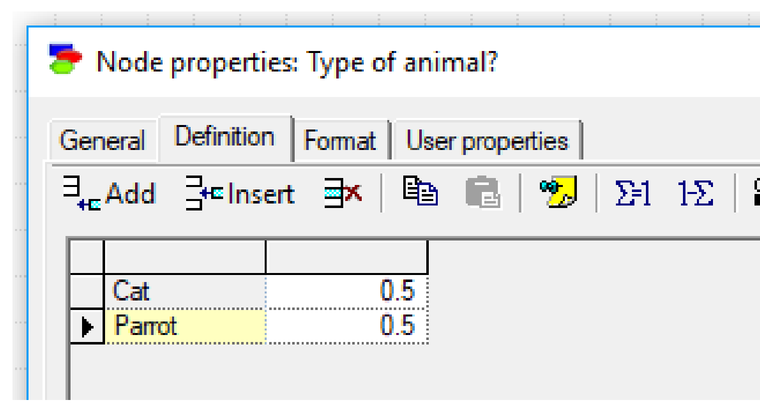



State identifiers, can be altered using the same format as for the node identifier (start with letter, use letters, digits and underscores)

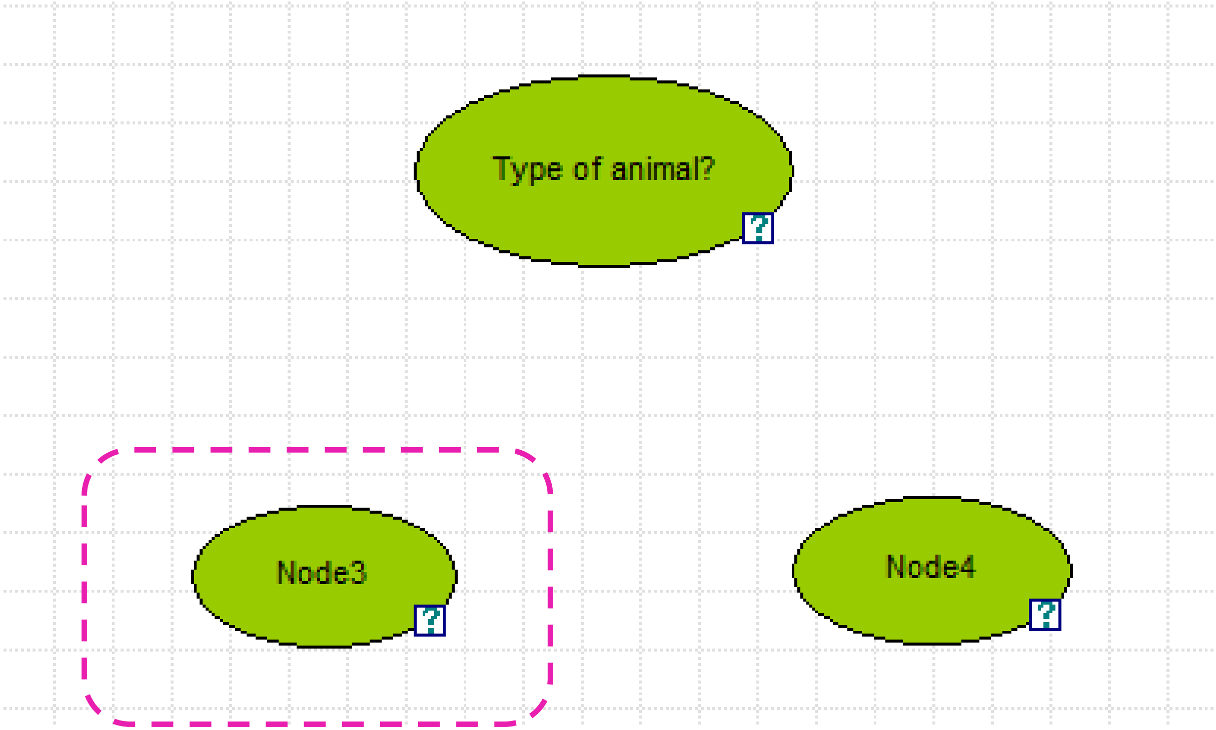
State probabilities (to be assigned). Since this node has (yet) no parents, the probabilities are unconditional (*on other information than background information*).

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

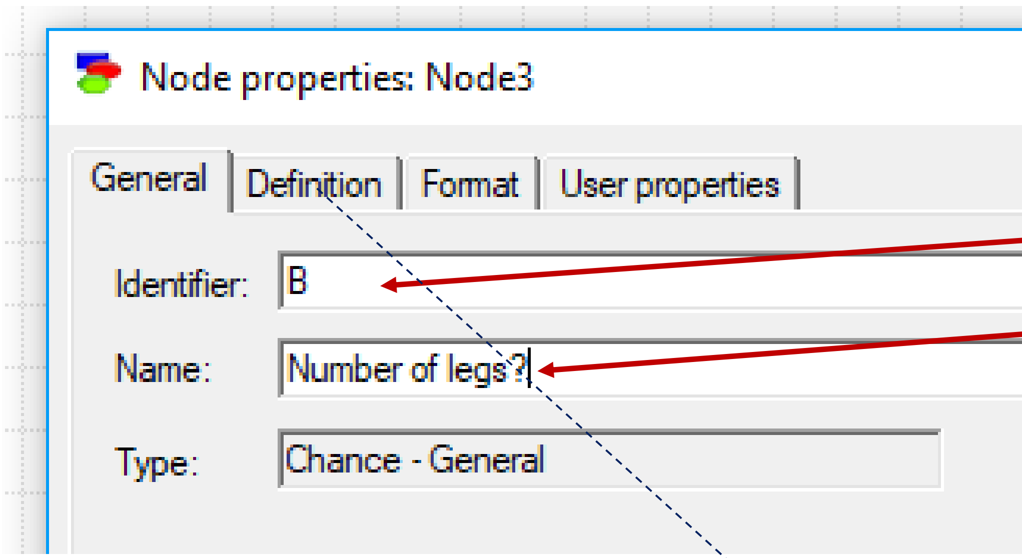
for instance…

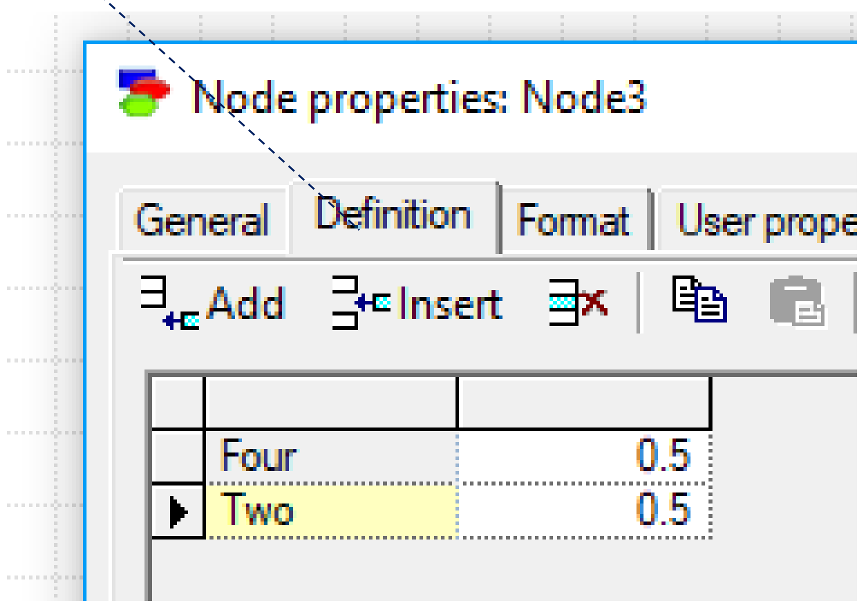


Add two more chance nodes…

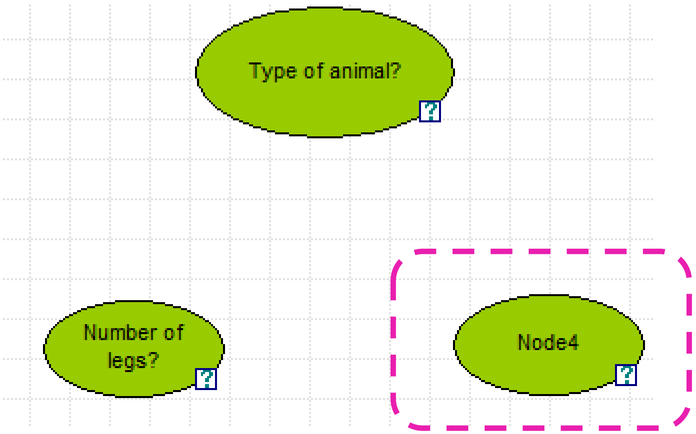


$B_1$ = “Willie has four legs”
$B_2$ = “Willie has two legs”

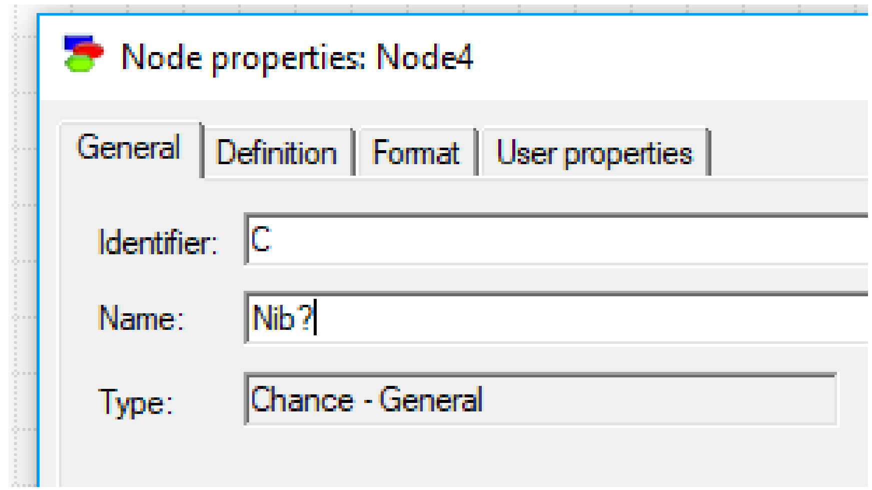


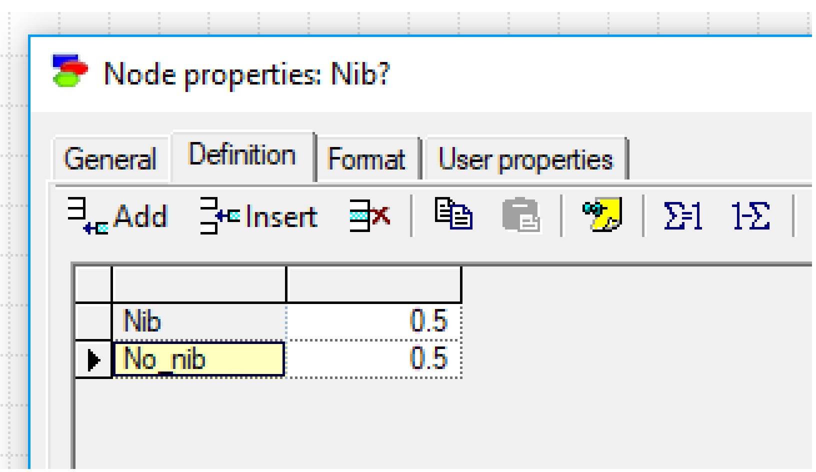




At this moment, do not bother about the probabilities.



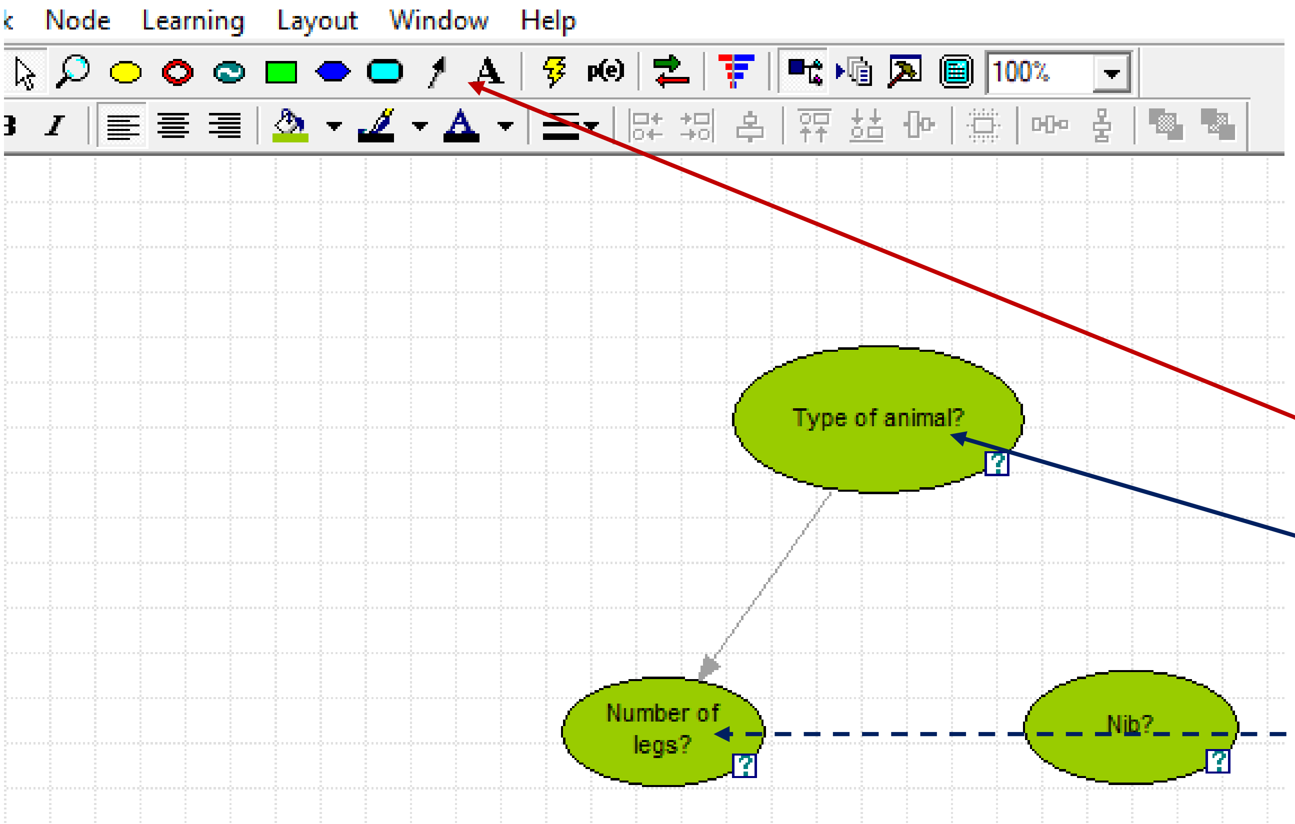

$C_1$ = "Willie has a nib"
$C_2$ = "Willie has no nib"




Note the format. Spaces are not allowed in state identifiers, use underscores.

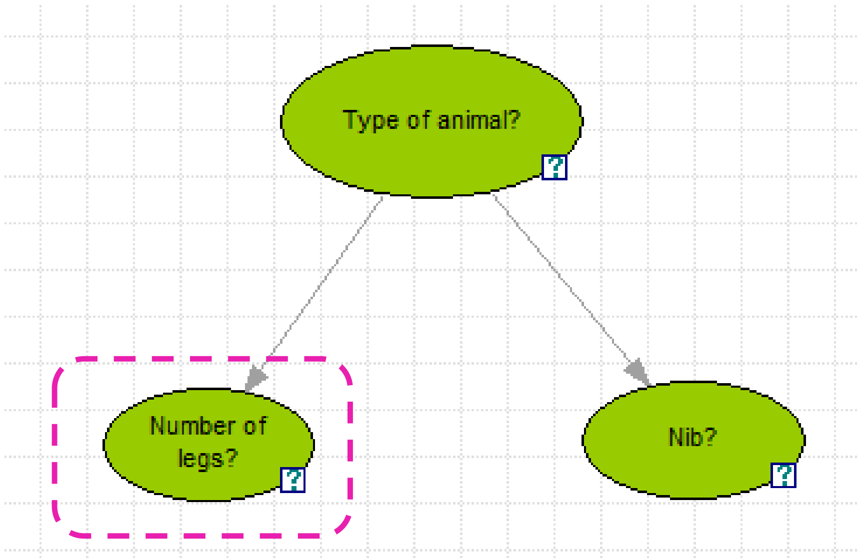
# Add links (edges)…




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.



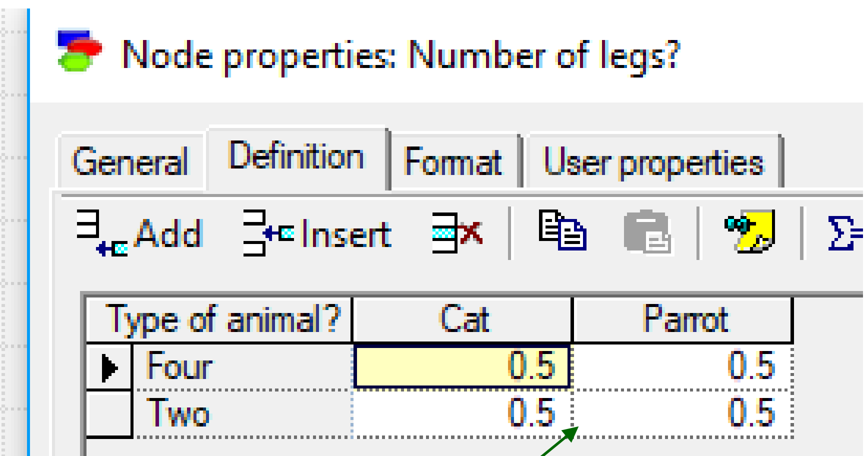
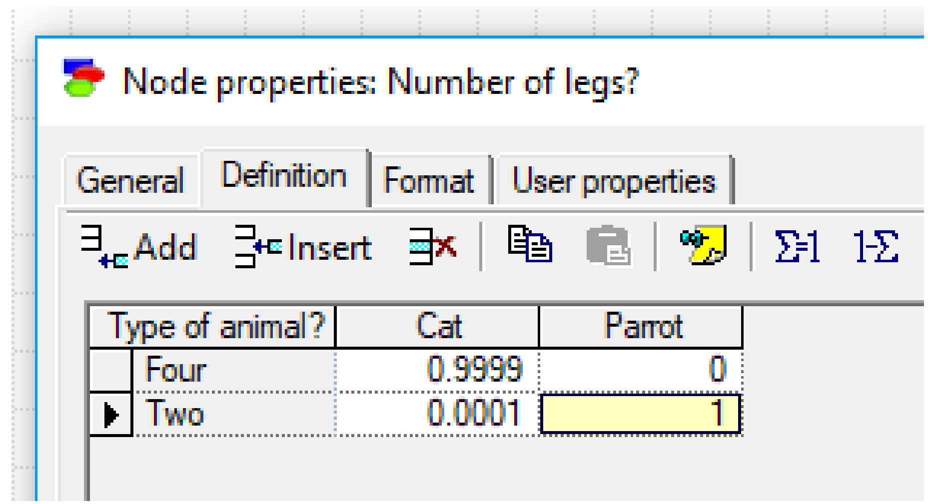
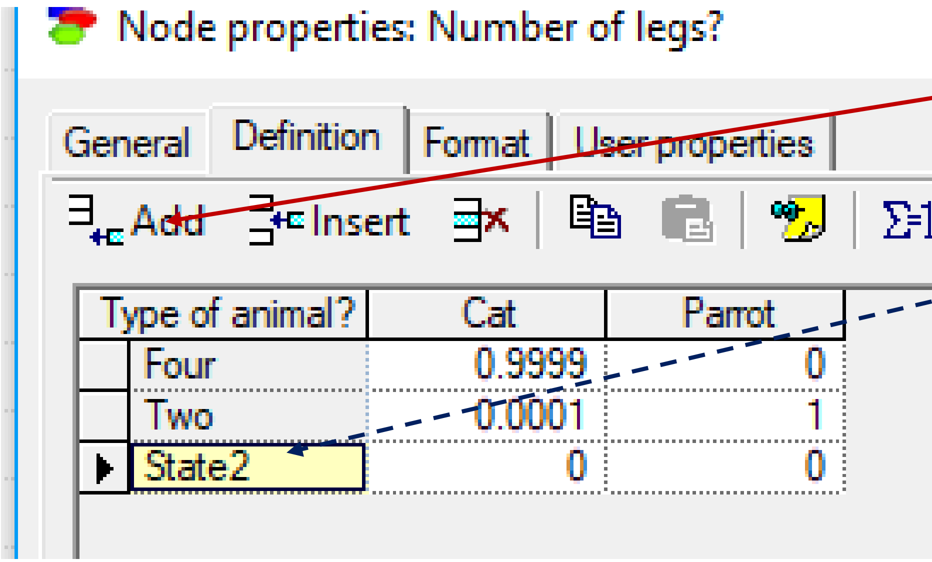




Table with probabilities of states (number of legs) conditional on state of parent node (type of animal)



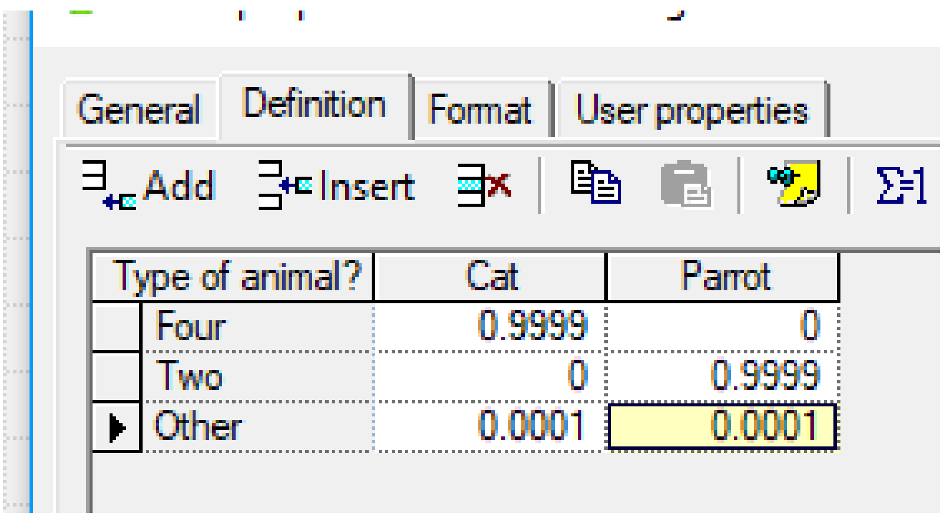

Reasonable settings?

# Add states…

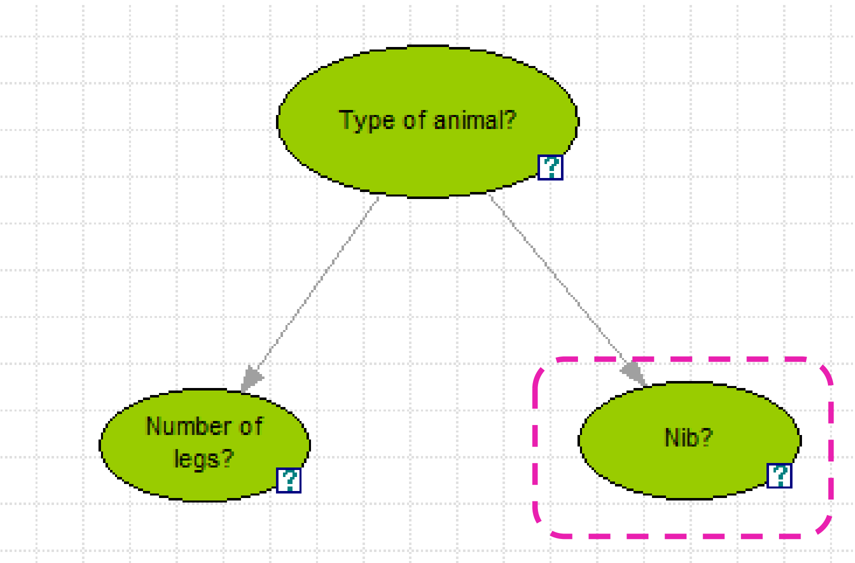


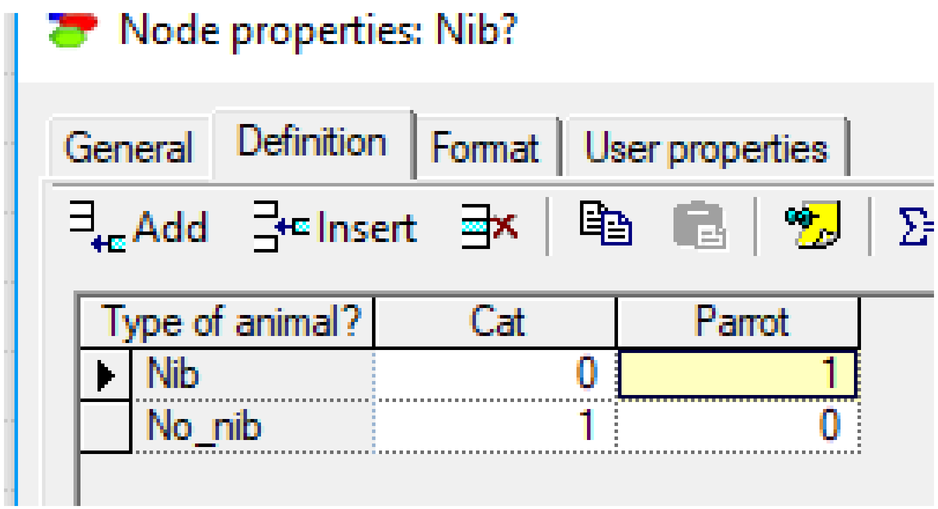

Click here

…and a new state is added




Settings more reasonable?

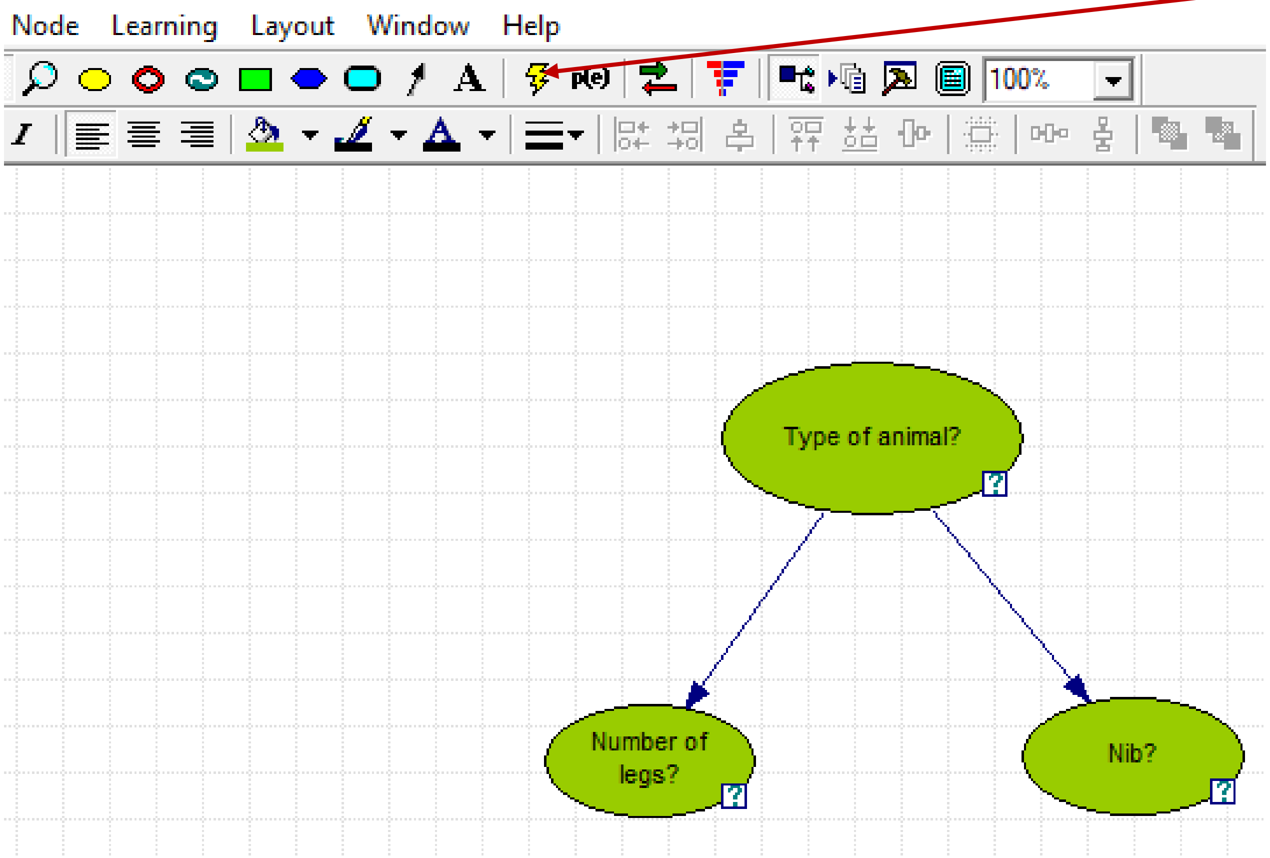


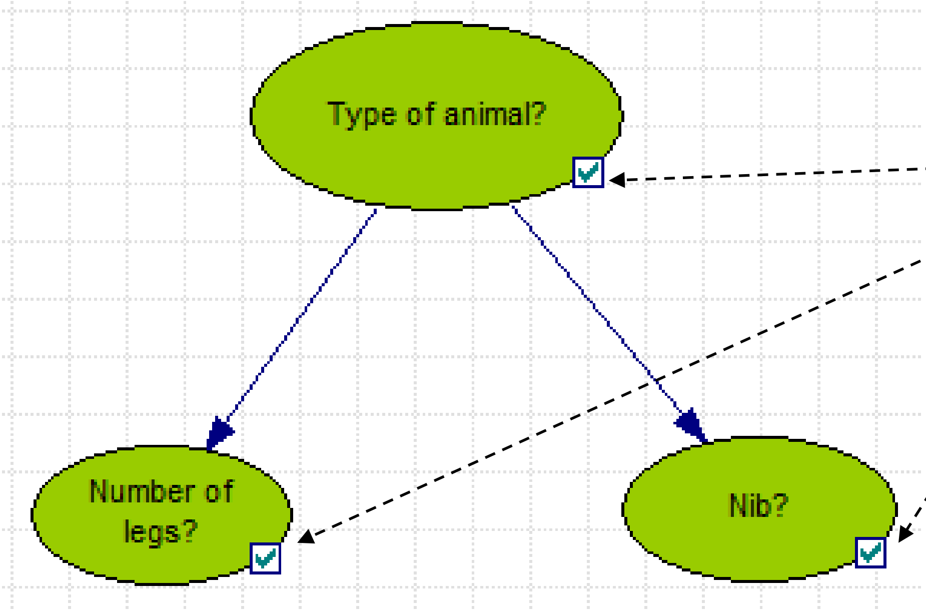




Reasonable settings?

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

Click on the flash symbol

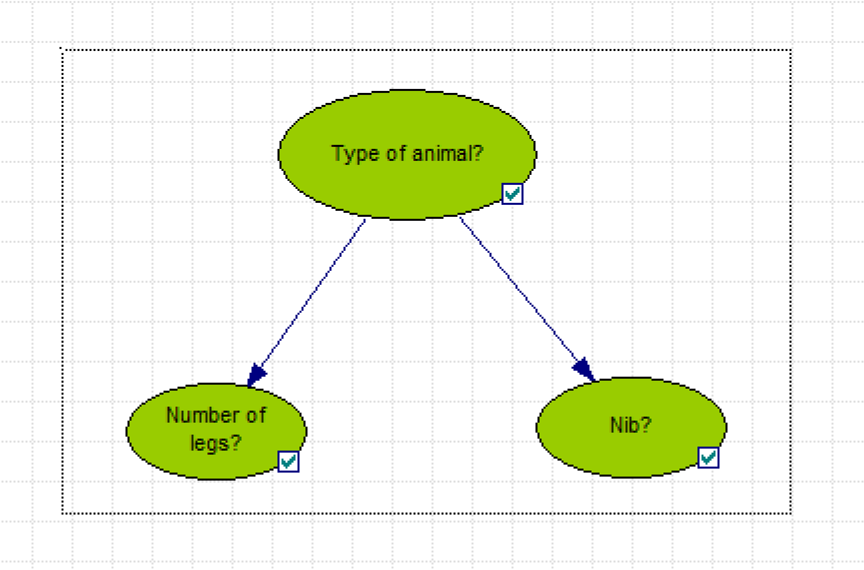


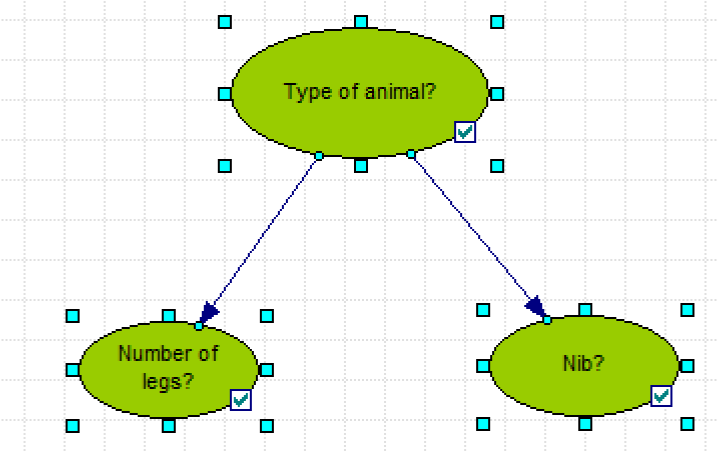



Check boxes have turned from showing a question mark (cf. previous slide) to showing the symbol for “checked”.

Upon running every node should show calculated marginal probabilities.
How can we see these?

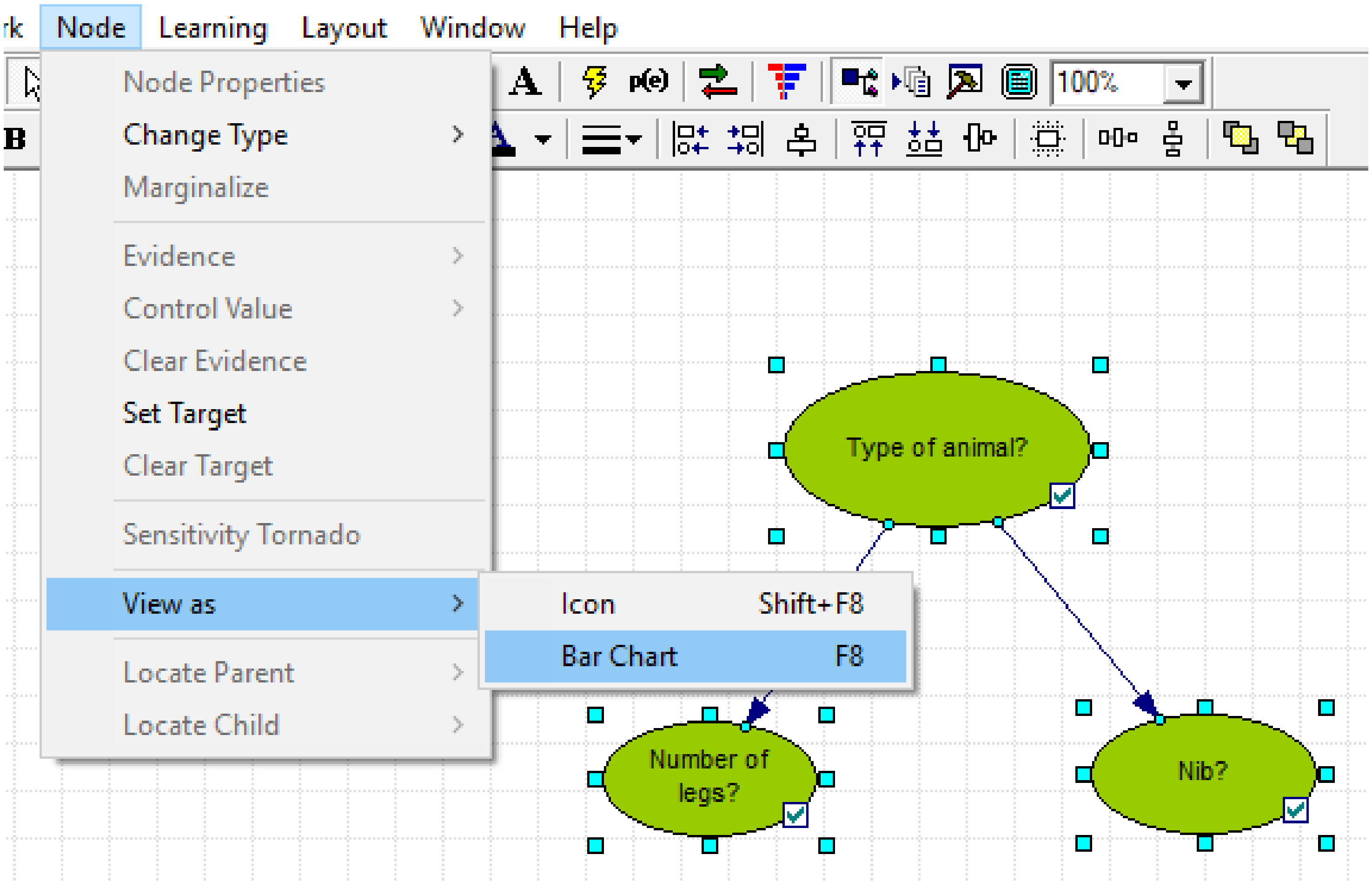
Click and drag the pointer so the dragging rectangle covers all three nodes

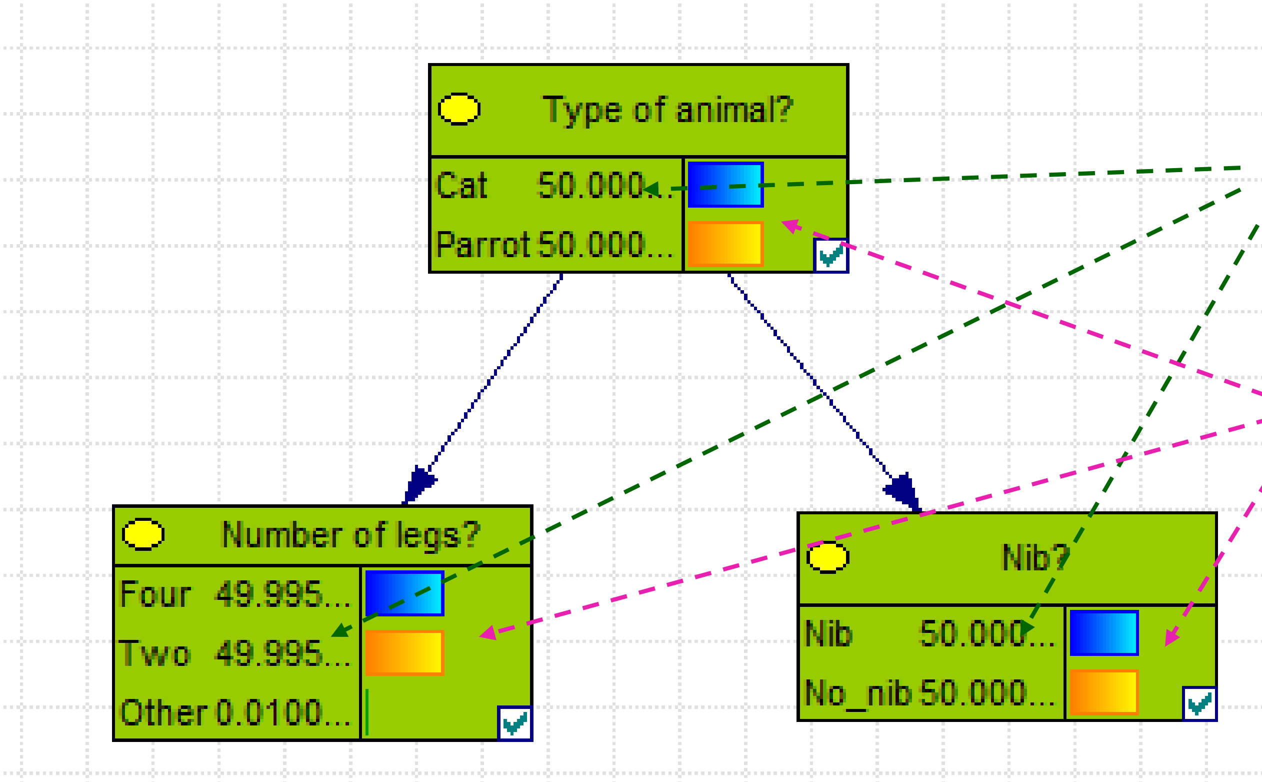



The green squares around a node indicates that the node is selected. Hence, all nodes are selected by this action.

Open the menu **Node** and select **View as** and then **Bar Chart**

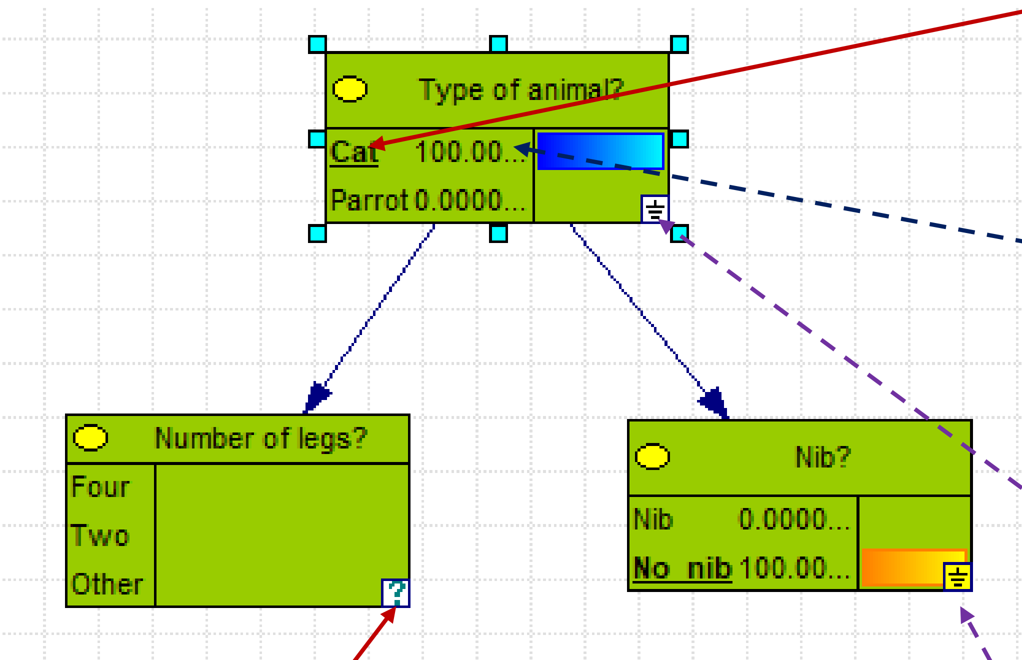





The numbers show the marginal probabilities expressed in percent.

The blue and yellow bars are just bar plots for these probabilities.

Now, we can see what happens when we fix a state in a node (*instantiating* the node), hence assuming temporarily that it is the true state (probability 100%).



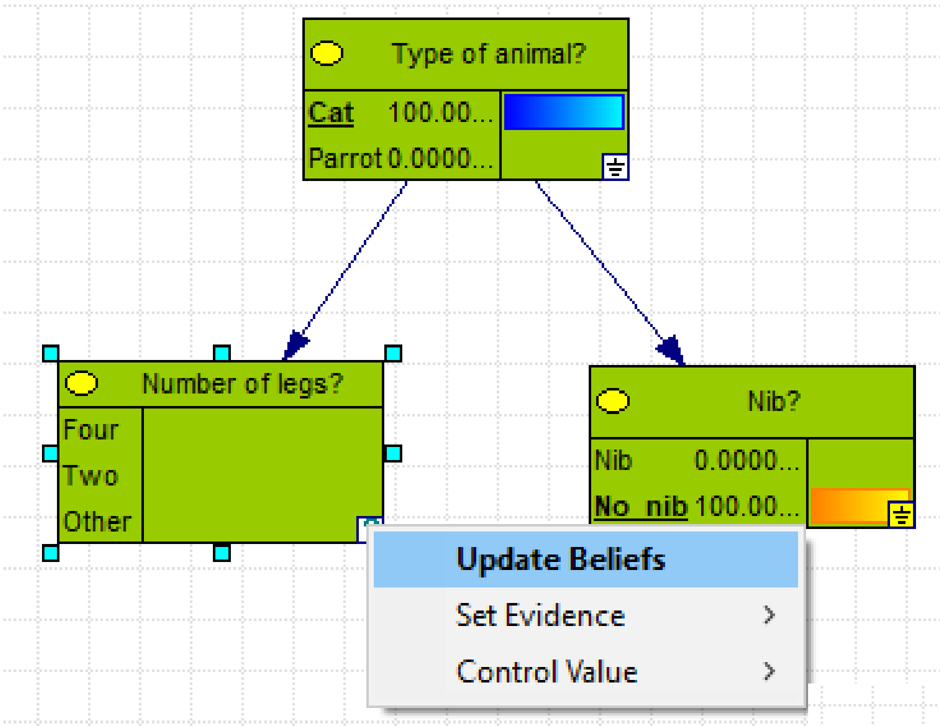

Double-click on the state "Cat" in node A.

The probability changes to 100%, the blue bar gets max length and the yellow bar disappears (0%).

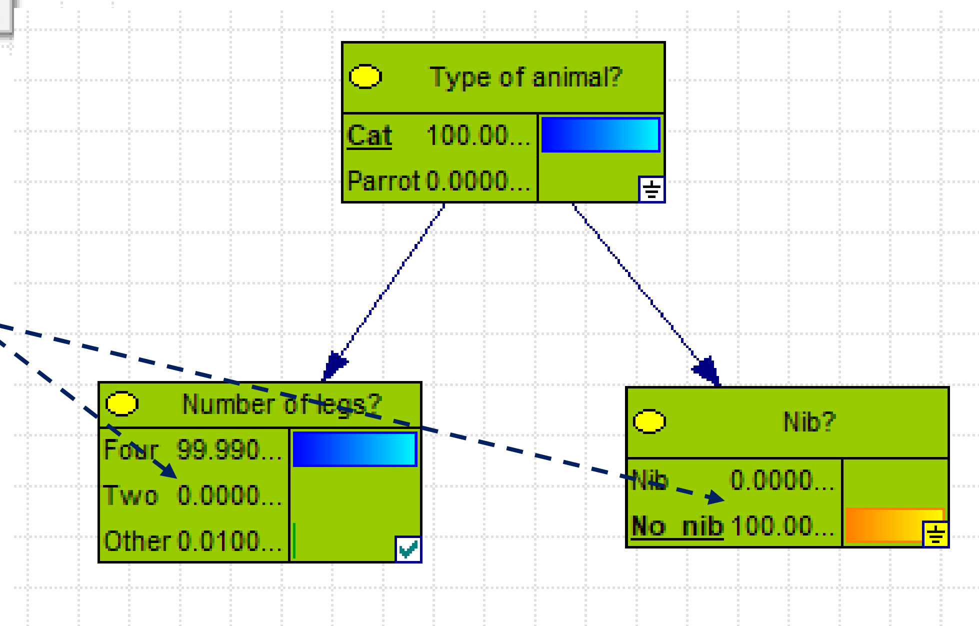
The check box here changes from showing symbol "checked" to symbol "earthed".

…and also here (why?)

Right-click on the question mark here.

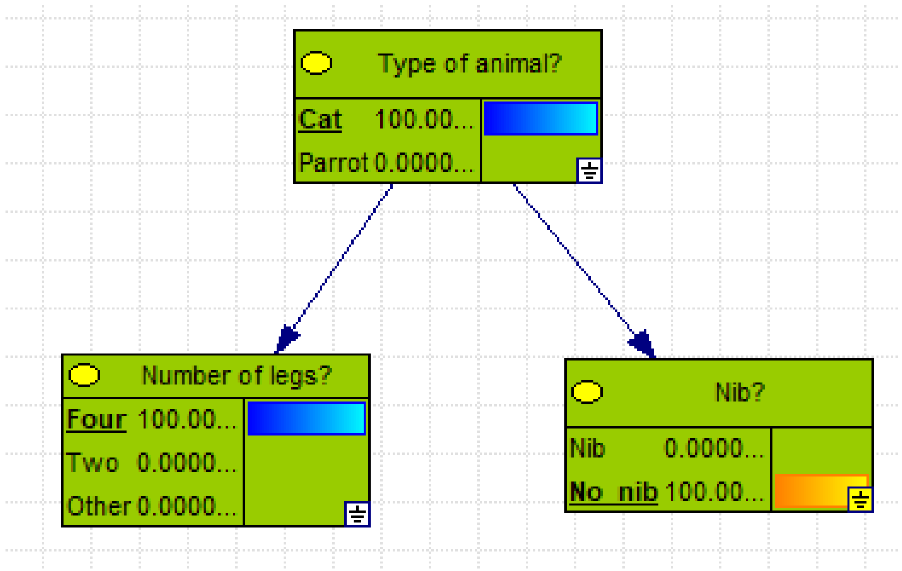



Select **Update Beliefs**



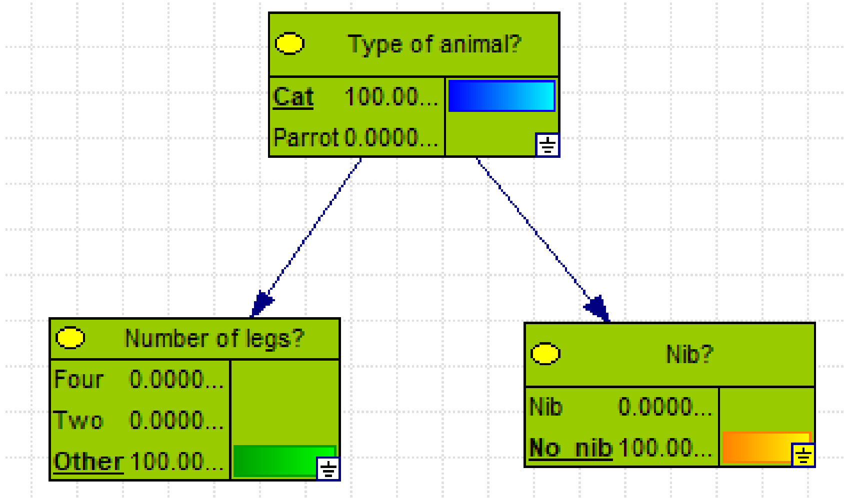

These are the conditional probabilities of the states in node B and node C given the state in node A is “Cat” (which we actually have assigned in the set-up of the network).

Now, we can try to instantiate node B to one of its states, for instance “Four legs”:




Nothing happens in node C

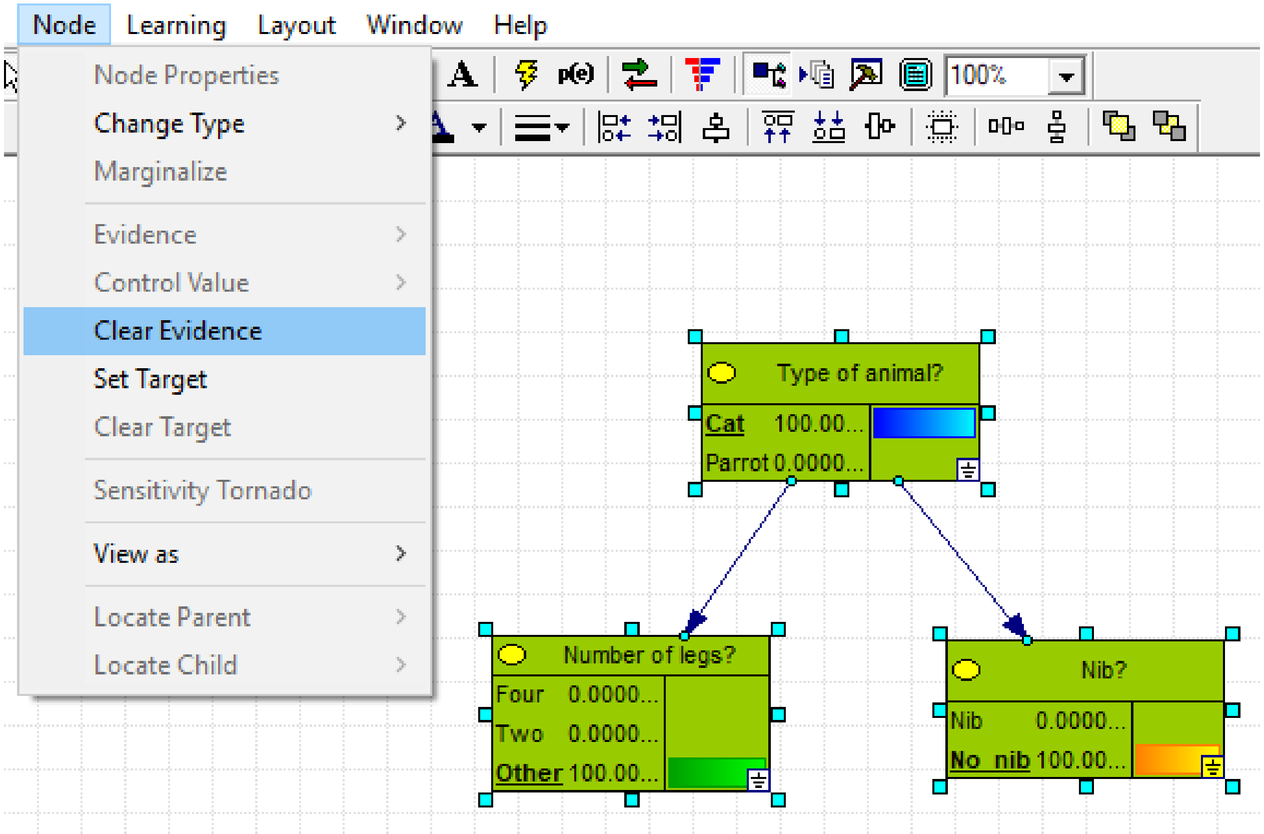
Try with state “Other”:



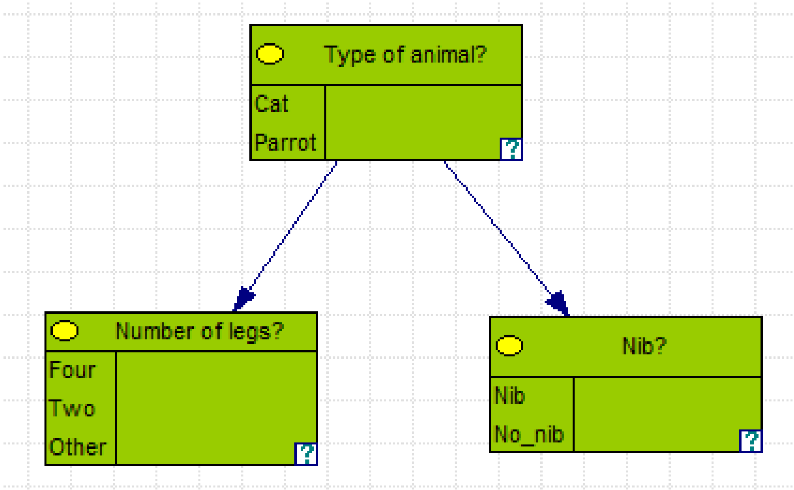

Still, nothing happens in node C

Given a state in node A, node B and node C are conditionally independent!

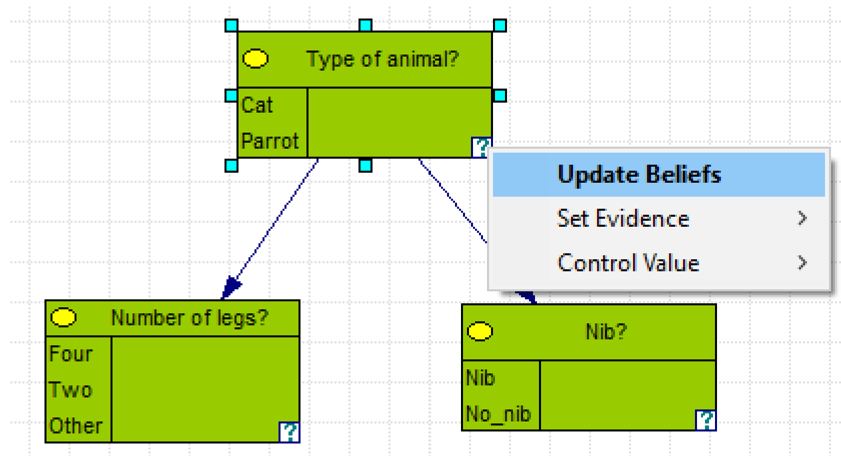
Now, we shall reset the network, i.e. clear all evidence.

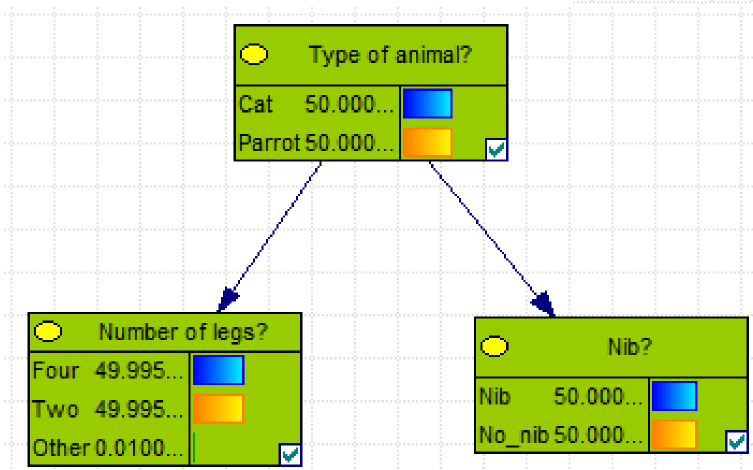



Select all nodes (click and drag the pointer over them) and select **Clear evidence** from the menu **Node**.



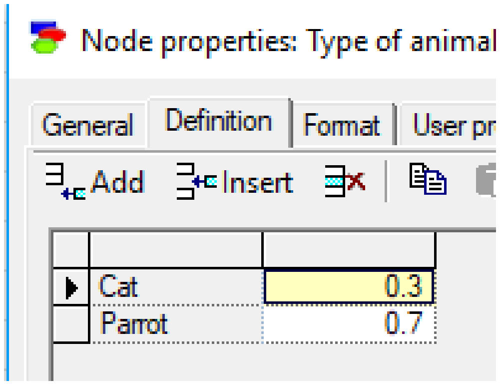

Complete the resetting by right-clicking on one of the question marks and select **Update Beliefs**



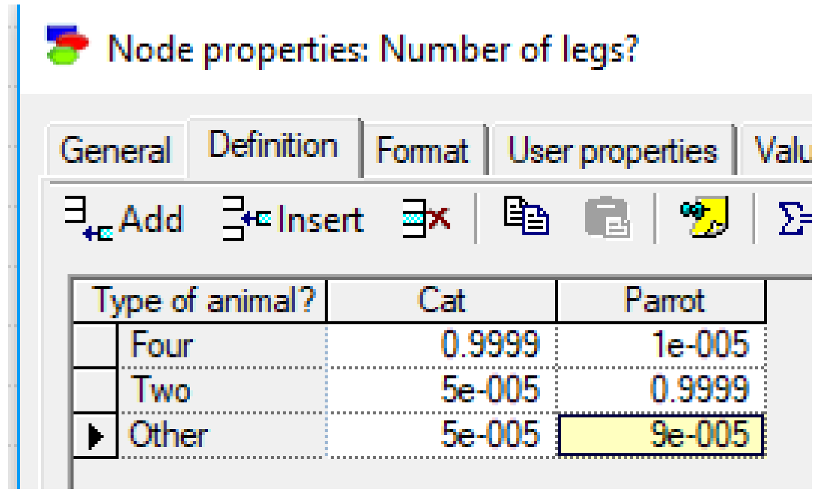



For illustration, change the probability settings (even if they become very strange):

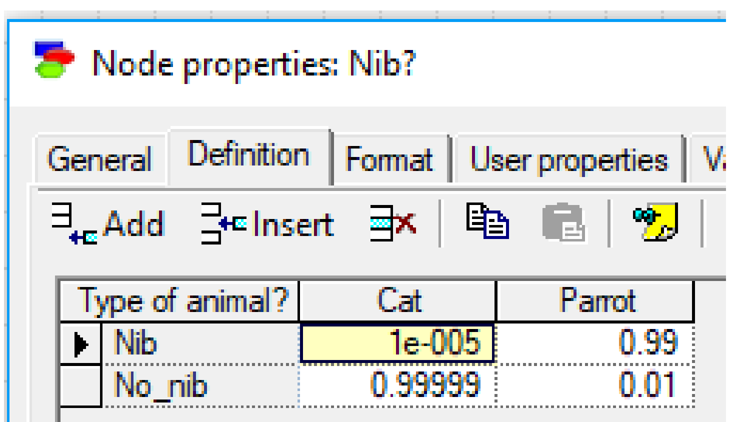
Node A

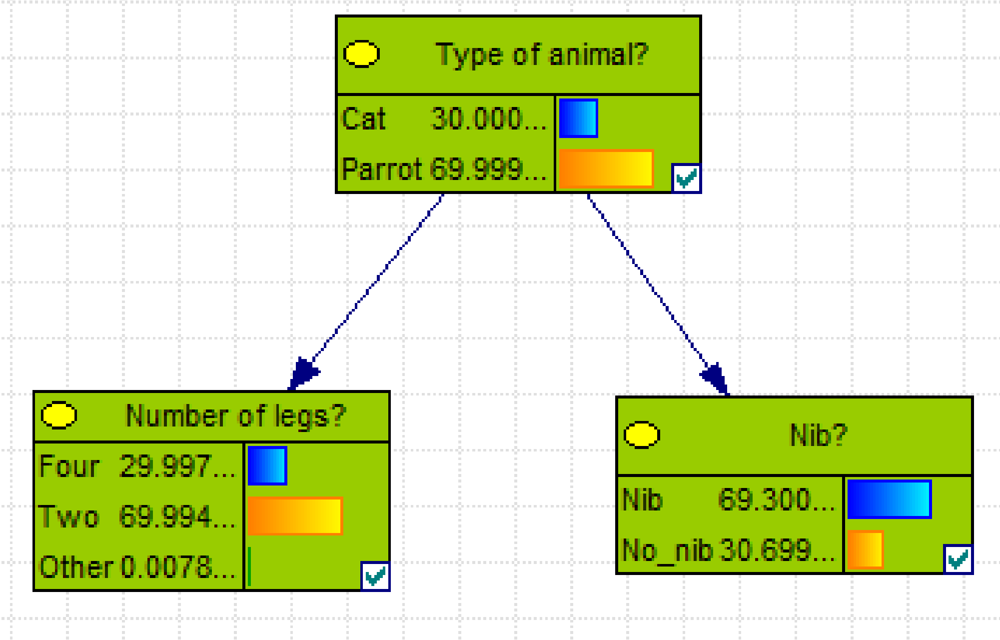



Node B

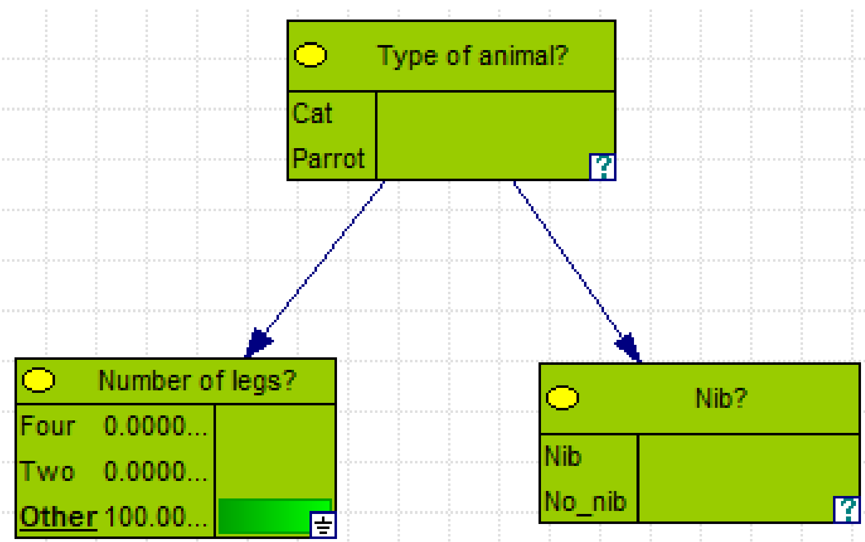



Node C

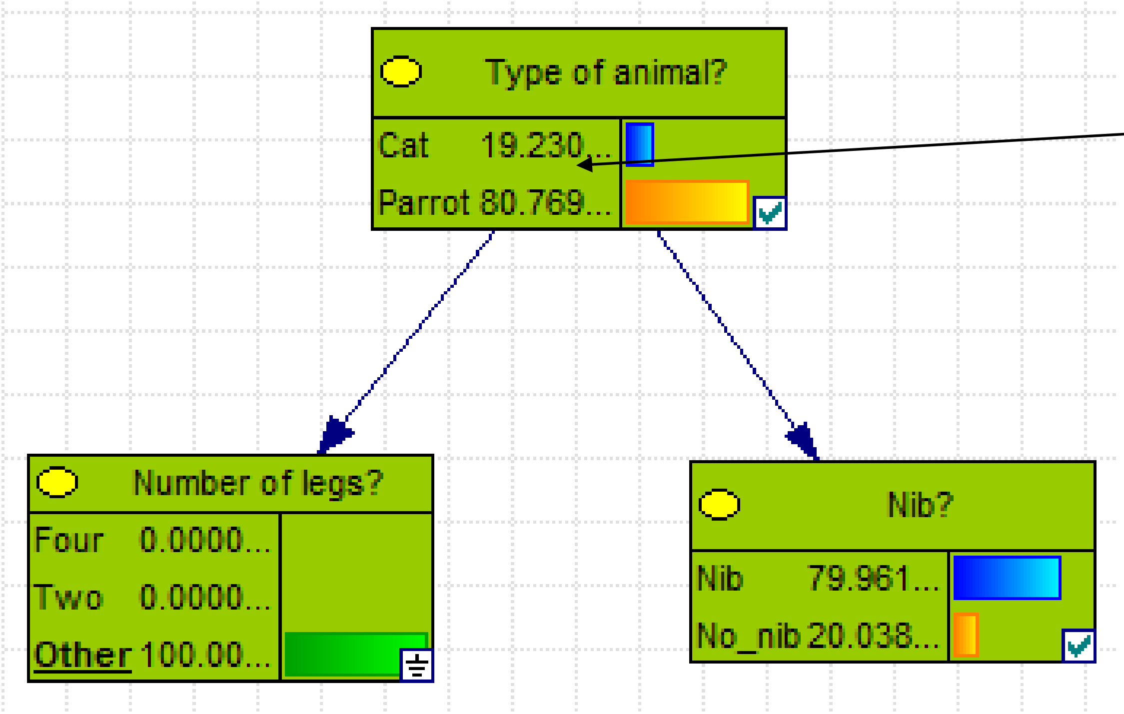





What will happen if we instantiate one of nodes B or C to one state?
For instance to state “Other” in node B:

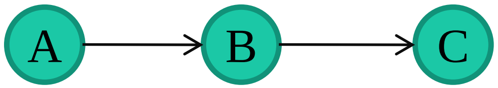



Complete by right-clicking on one of the question marks and select **Update Beliefs**




These probabilities are now the (computed) conditional probabilities of the states in node A <u>given</u> the state “Other” in node B.

2) Serial connection



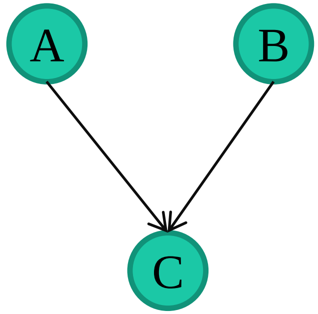

There is a path between  A and C (unidirectional from A to C).

➔ A may be relevant for C (and vice versa).

If the state of B is known this relevance is lost.: The path is *blocked*.

➔ A and C are *conditionally independent* given (a state of) B.

3) Converging connection




There is a path between A and B (not unidirectional).

➔ A may be relevant for B (and vice versa).

If the state of C is (completely) unknown this relevance does not exist.

If the state of C is known (exactly or by a modification of the state probabilities) the path is opened.

➔ A and C are *conditionally dependent* given information about the states of C, otherwise they are (conditionally) independent.

*Example* Paternity testing: child, mother and the true father

Let

A be a random variable representing the mother's *genotype* in a specific *locus*.

B be a random variable representing the true father's genotype in the same locus.

C be a random variable representing the child's genotype in that locus.

| | | |
|---|---|---|
| A: | $A_1$ | $A_2$ |
| B: | $B_1$ | $B_2$ |
| C: | $C_1$ | $C_2$ |

| | | |
|---|---|---|
| A: | $A_1$ | $A_2$ |
| B: | $B_1$ | $B_2$ |
| C: | $C_1$ | $C_2$ |

A: Mother's genotype
B: True father's genotype
C: Child's genotype

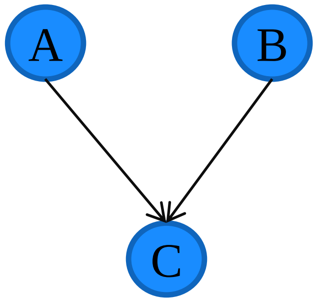
If we know nothing about C ($C_1$ and $C_2$ are both unknown) , then

- information about A cannot have any impact on B and vice versa

If we on the other hand know the genotype of the child ($C_1$ and $C_2$ are both known or one of them is) then

- knowledge of the genotype of the mother has impact on the probabilities of the different genotypes that can be possessed by the true father since the child must have inherited half of the genotype from the mother and the other half from the father

Bayesian network:



# 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*)**

Return to the example with banknotes. Let

$H_0$: Dye is present

$H_1$: Dye is not present

*States of nature*

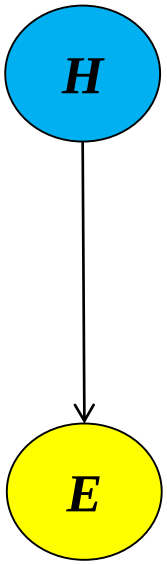
and let

$E_1$: Method gives positive detection

$E_2$: Method gives negative detection

*Data*

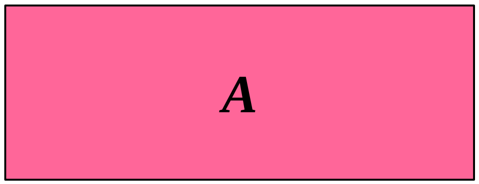
A simple Bayesian network can be constructed for the relevance between the state of nature and data:



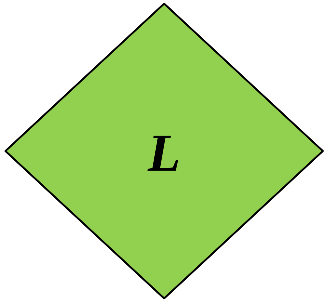

| ***H*** | *Probabilities* |
|---|---|
| $H_0$ | 0.001 |
| $H_1$ | 0.999 |

| | | *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 loss function



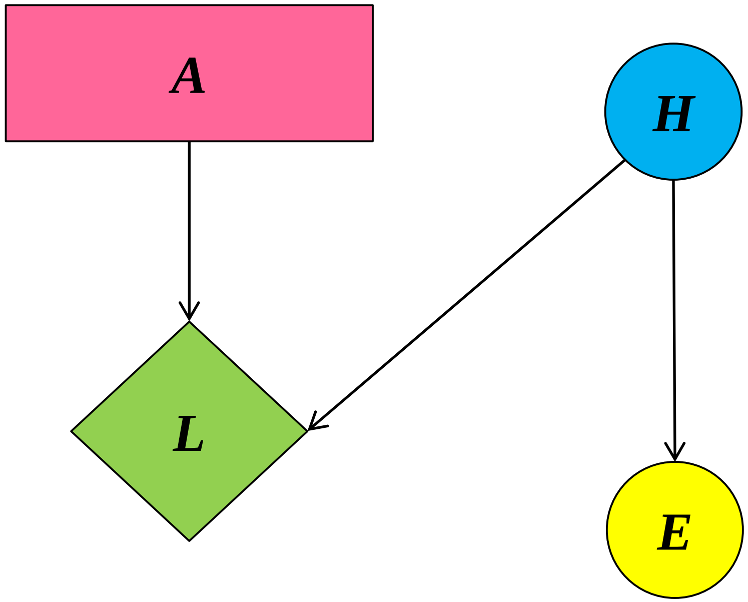

| $\boldsymbol{A}$ | |
|---|---|
| $a_1$ | Destroy banknote |
| $a_2$ | Use banknote |




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

Neither of the nodes are random nodes.

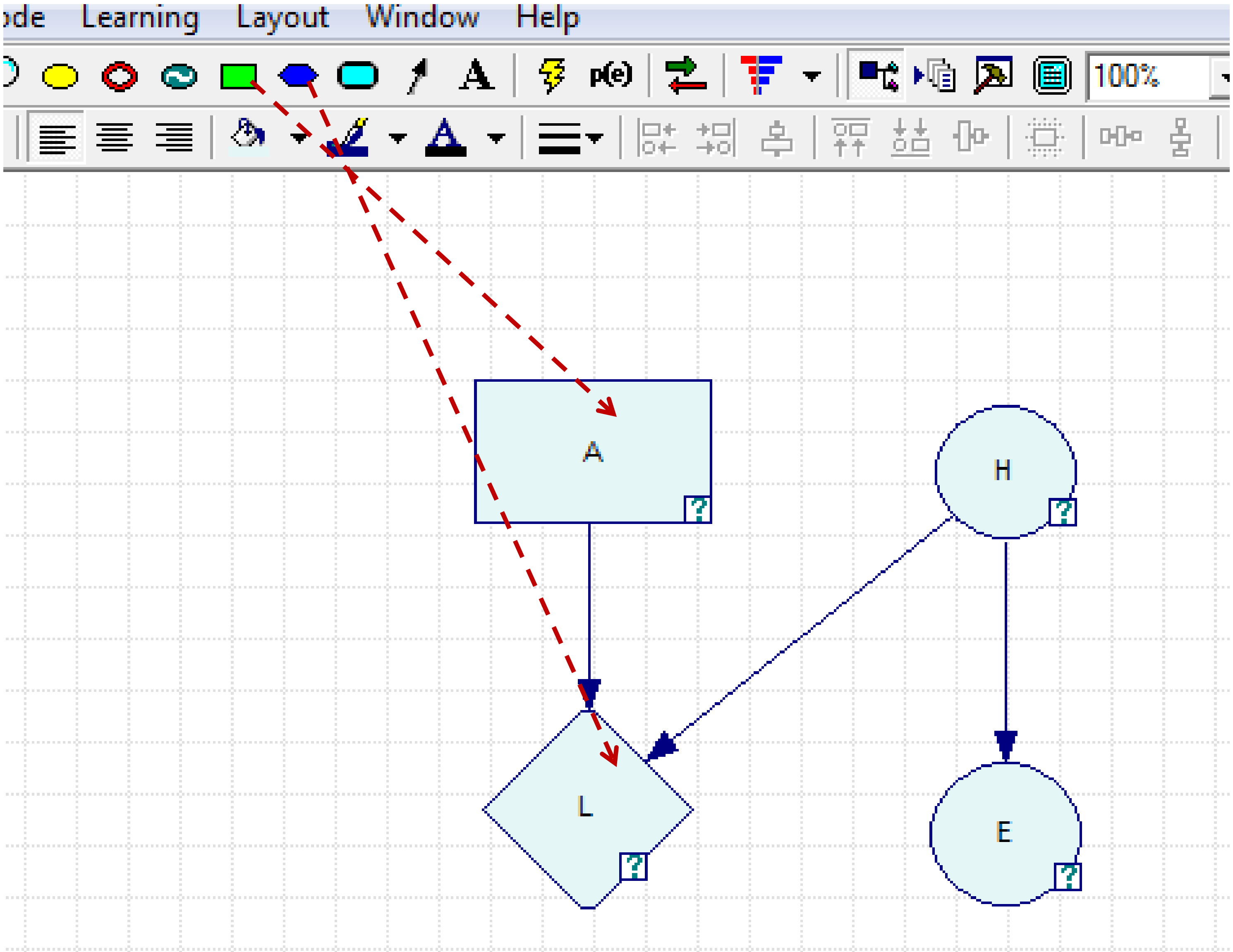
Node ***L*** must be a child node with nodes ***H*** and ***A*** as parents.

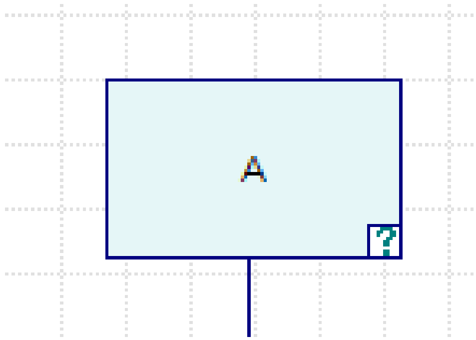


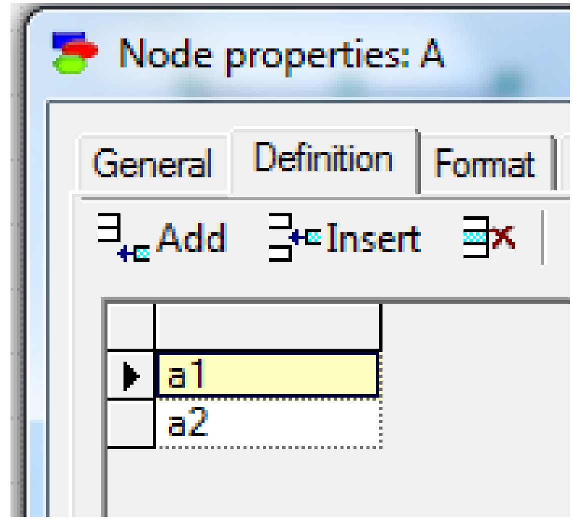

With this network, an influence diagram, we would like to be able to propagate data from node *E* to a choice of decision in node *A*.

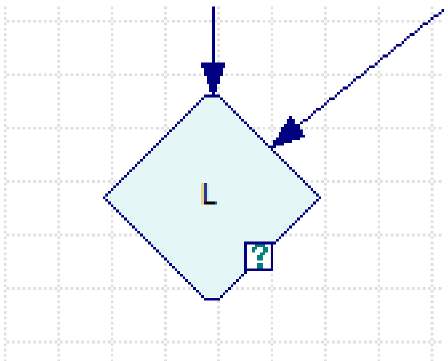
Hence, in the loss node *L* the expected posterior loss should be calculated.

Using GeNIe:

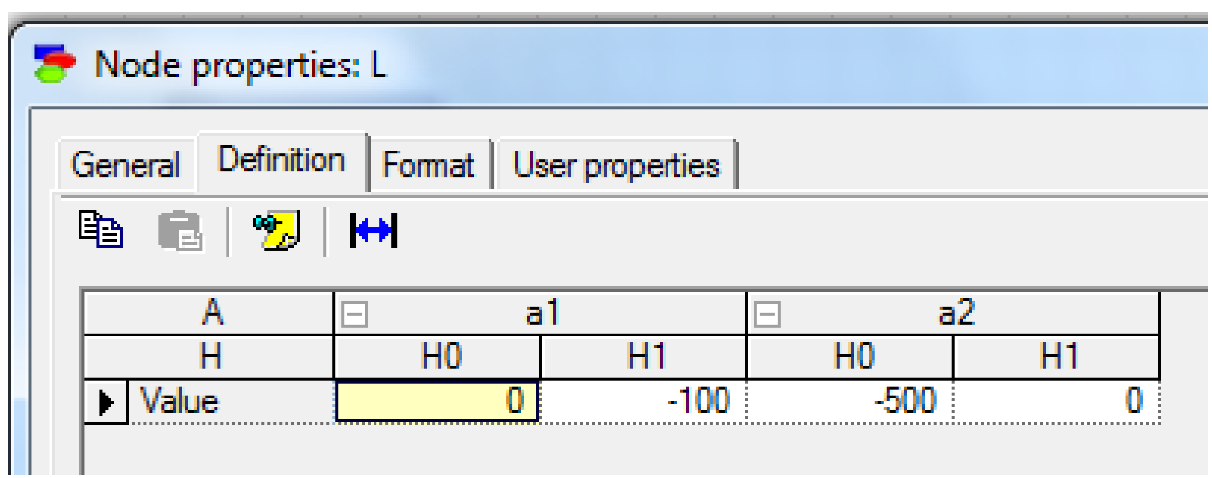




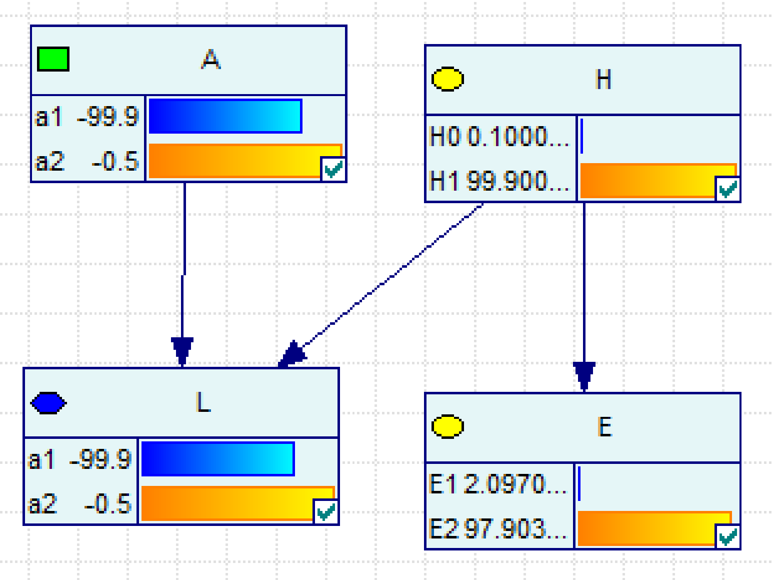



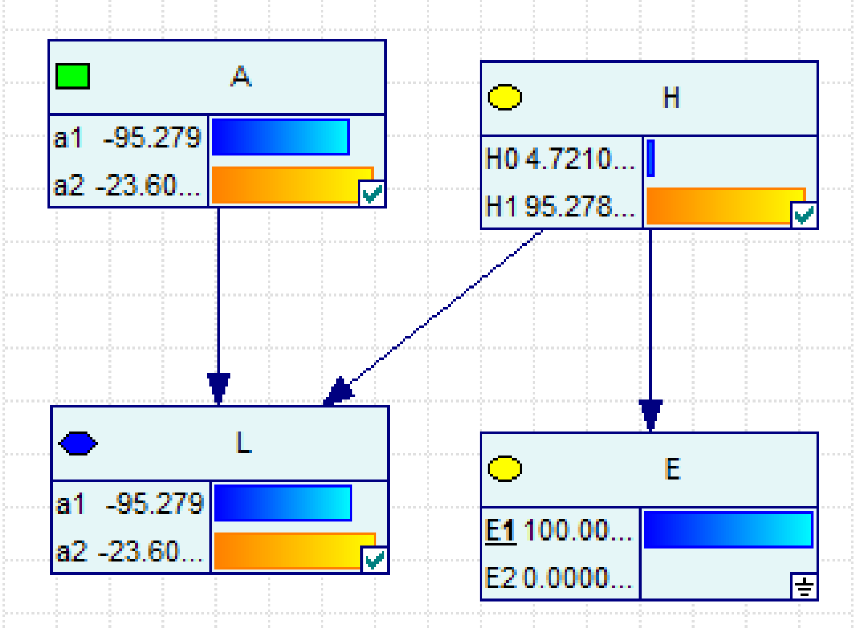

In GeNIe (and other software), this node is per definition a *utility node*, but it can be used as a loss node by representing losses as negative utilities



Run the network




Instantiate node ***E*** to $E_1$




The expected posterior utility (negative loss) can be read off in node ***A*** (and also in node ***L*** )