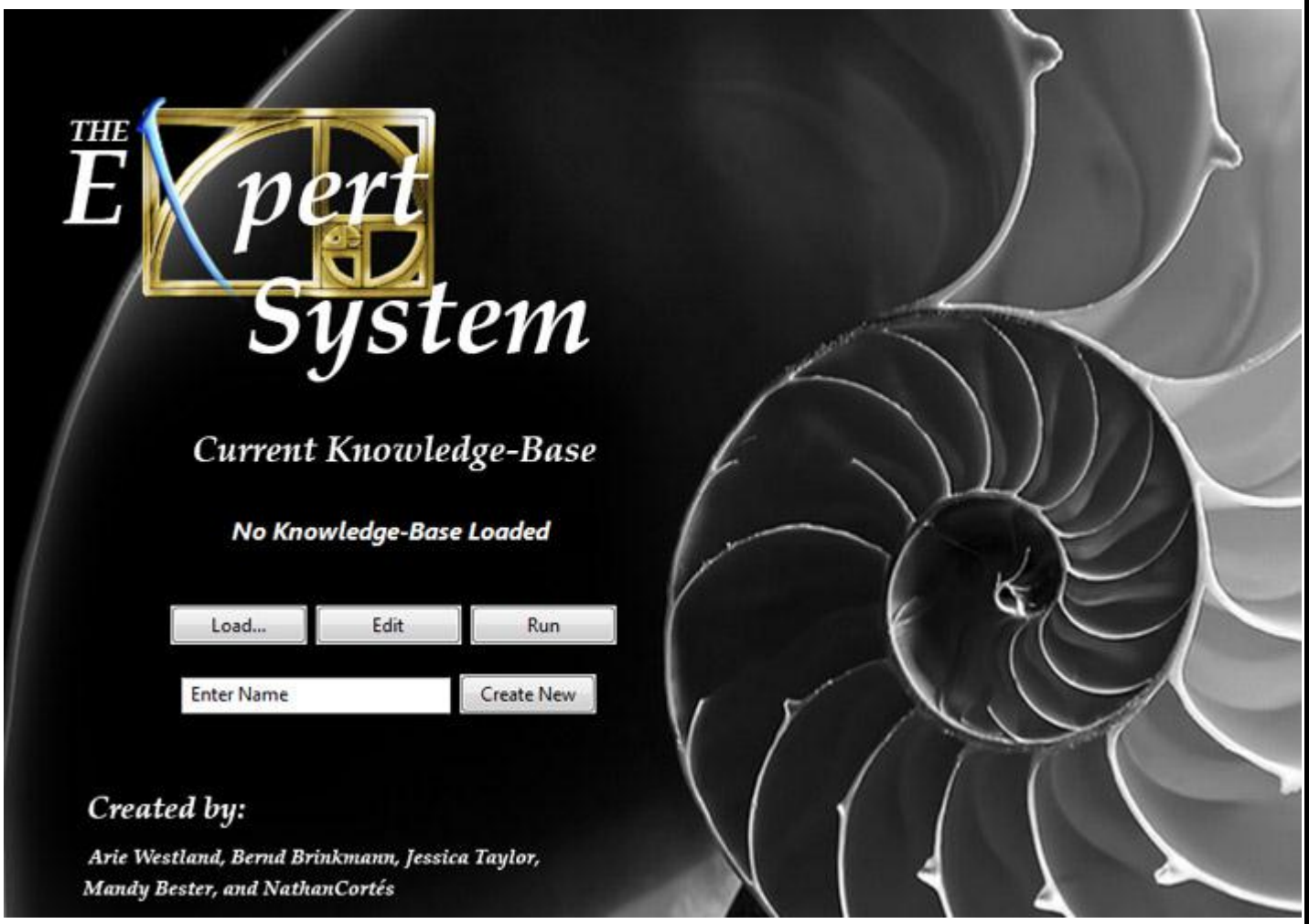
University of Tasmania, 2014

KNE441: COMPUTATIONAL INTELLIGENCE

EXPERT SYSTEM SHELL



User Manual

Creators:

Arie Westland, Bernd Brinkmann, Jessica Taylor, Mandy Bester, Nathan Cortés.

TABLE OF CONTENTS

INTRODUCTION

This product is an Expert System Shell called “The Expert System Shell”. Its purpose is to aid in creating Intelligent Systems, and simplify the development process. The user can run expert systems already stored in the program without needing to access the associated knowledge bases, whilst developers are able to create and store complex expert systems using a simple user interface.

The Expert System Shell supports the use of both Forward Chaining and Backward Chaining inference algorithms for computation, and provides the option of implementing Bayesian Reasoning and Certainty factor techniques to facilitate for any uncertainty in data. Conflict Resolution can also be implemented using this product using the most specific technique. This technique reorders the rules so that the consequent of the most specific rule able to be fired is chosen as the answer. It is important to note that the results of an evaluation using this technique will depend on whether backward chaining or forward chaining is selected as a run method.

As there are two different ways in which people may wish to use this product, two interfaces have been developed. The *Run Knowledge-Base* interface can run an already stored Knowledge-Base. The user can select a run method, i.e. Forward Chaining, Backward Chaining, or Default (which implements forward chaining and is included for users that are new to artificial intelligence); and can additionally select a reasoning method, i.e. Bayesian Reasoning and Certainty Factor techniques to allow for uncertainty in user input, or Default when there is no uncertainty. In the *Run Knowledge-Base* interface you will be able to utilise the “How” and “Why” functions to help the user understand how a conclusion has been reached or why a particular piece of information is required, respectively. In the *Run Knowledge-Base* interface the Knowledge-Base is inaccessible, and thus changes cannot be made to it. The purpose of this mode is to separate the run Knowledge-Base and edit Knowledge-Base features of the program to ensure The Expert System Shell is as simple as possible for the user. The default settings for the *Run Knowledge-Base* interface is forward chaining with all other tools set to default.

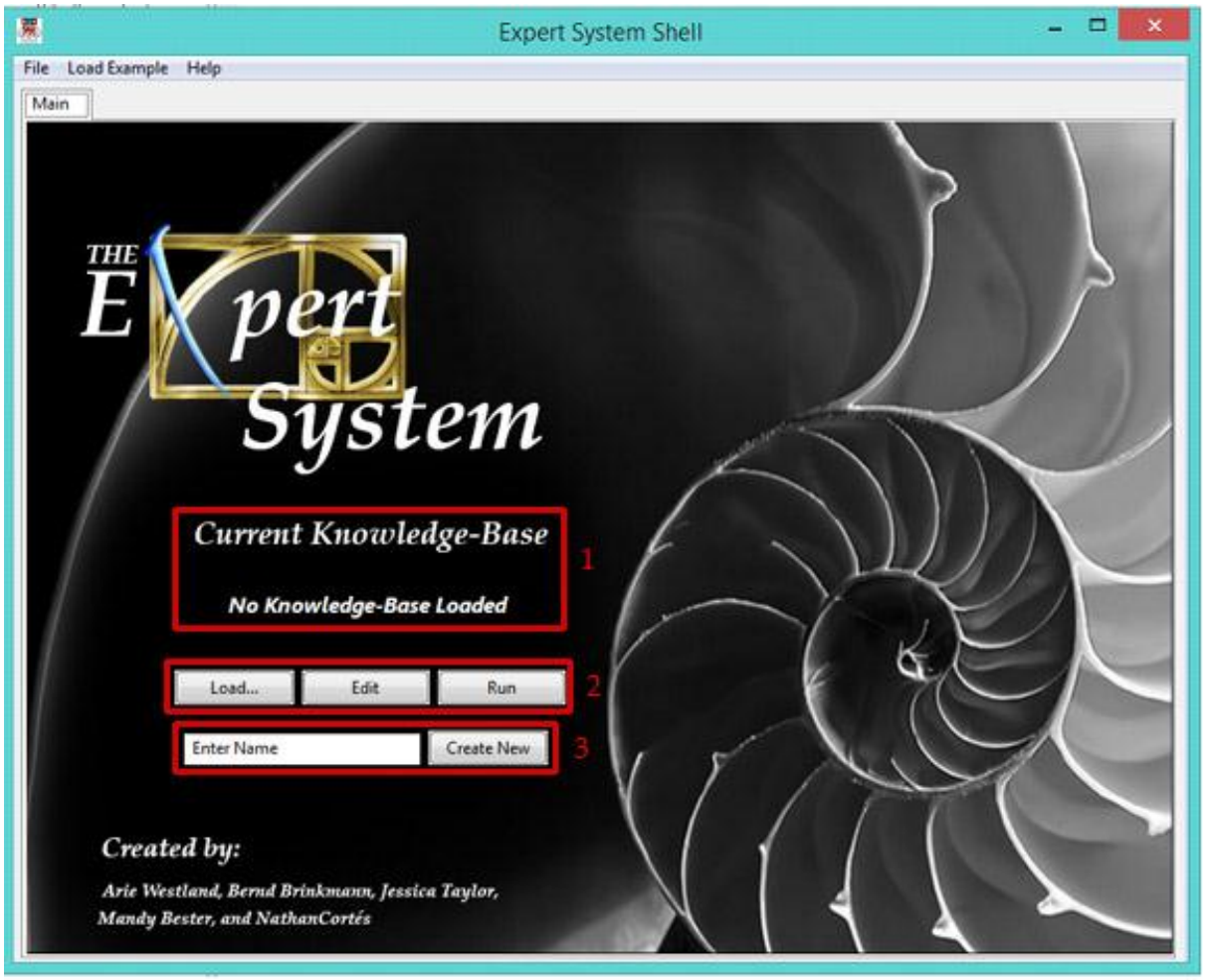
The *Create/Edit Knowledge-Base* interface has the ability to create a new Knowledge-Base and edit an existing Knowledge-Base. The user cannot run the Knowledge-Base in this mode, and must switch to the *Run Knowledge-Base* interface to do this. The *Create/Edit Knowledge-Base* interface allows the user to add rules, delete rules, and re-order rules. The user can additionally edit existing rules by adding/deleting/altering antecedents and consequents, and by setting respective values for the uncertainty methods if required.

The user manual will step though both the end user and developer interfaces, and describe the technical implementation of The Expert System Shell.

MAIN SCREEN

When you open The Expert System Shell the following Interface will appear in order to do anything with the program you will need to select one of the following options

- Do you wish to run an existing Knowledge-Base
- Or do you wish to create a new Knowledge-Base



1- This field tells you which Knowledge-Base you currently have loaded (in this case no Knowledge-Base is loaded).

2-

   a. 'Load…' loads an existing Knowledge-Base. Clicking will take you to a directory to choose your Knowledge-Base from a .kb file type.
   b. 'Edit' lets you edit or create a Knowledge-Base select the edit button. This will open two tabs, both the Create/Edit Knowledge-Base tab and the Variables tab.
   c. Clicking the 'Run button will open the 'Run Knowledge-Base' tab. From here you can run the selected Knowledge-Base.

3- To create a new Knowledge-Base, enter the name of the Knowledge-Base into this field and click the 'Create New' button.

MENU

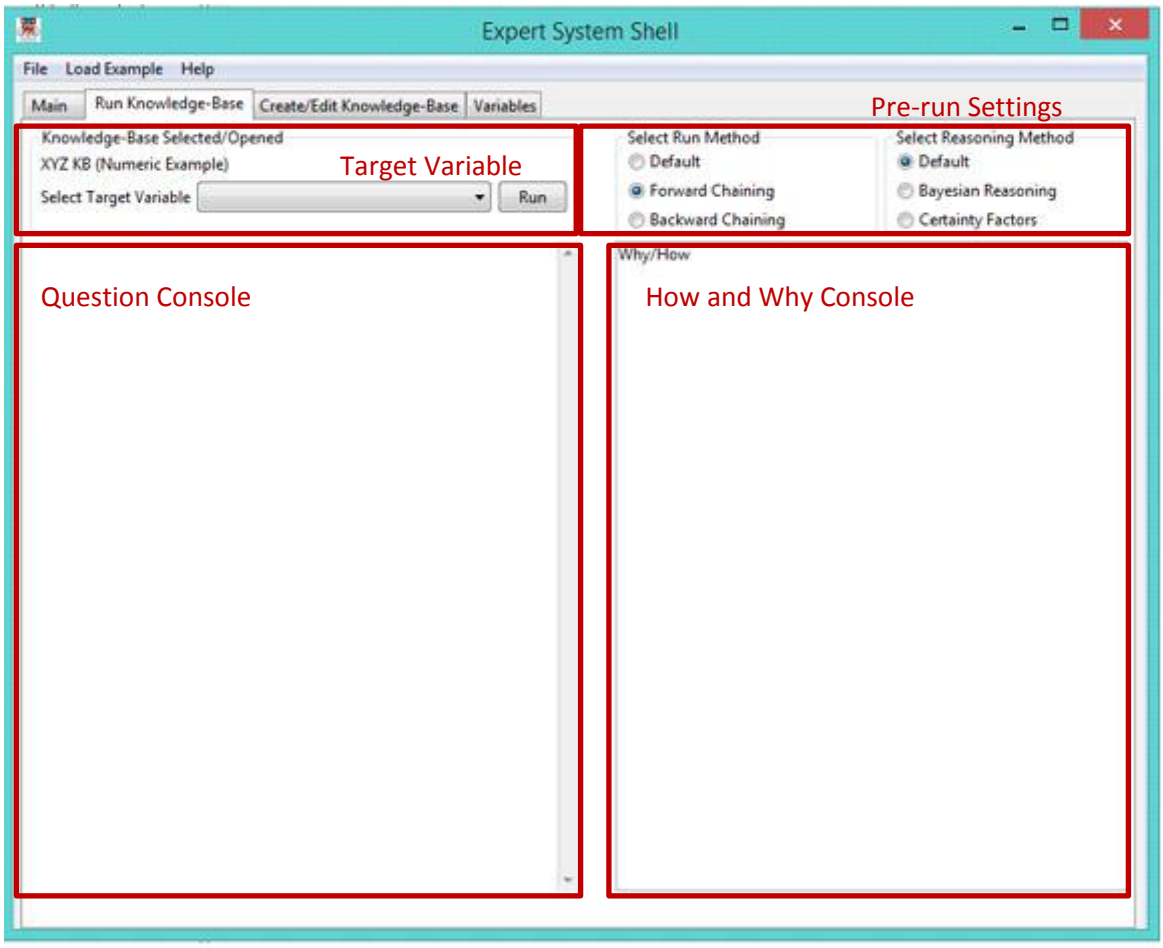
The menu allows access to several functions of the program.

- The ‘File’ menu item allows a user to load or save a Knowledge-Base.
- The ‘Load Example’ menu item lets the user load in a hard-coded example Knowledge-Base.
- The ‘Help’ menu item gives the options of opening the ‘Quick-start Guide’, ‘User Manual’ or ‘About’ screens.

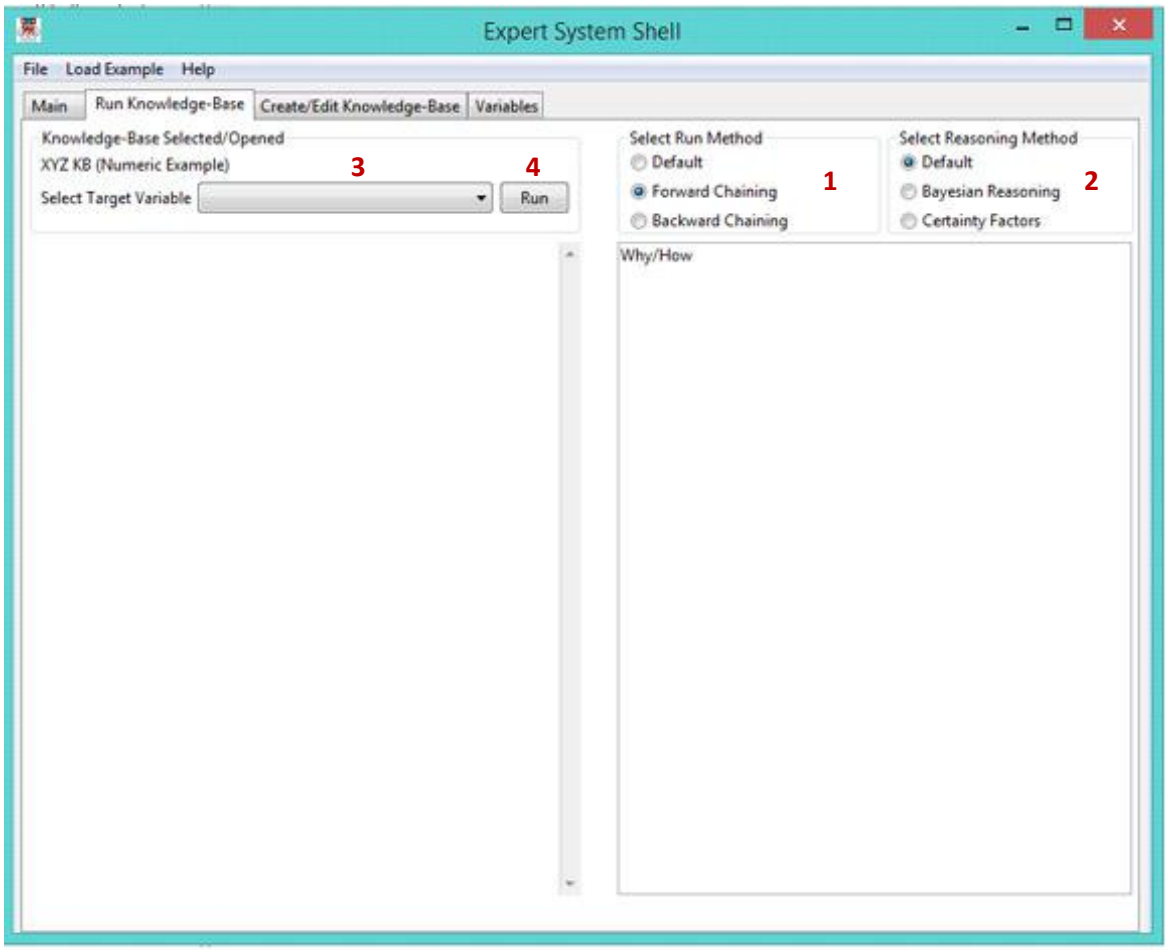
RUN KNOWLEDGE-BASE INTERFACE

The Run Knowledge-Base Interface has features which make it simple for the user to run the Knowledge-Base without getting confused by the functions that enable experienced users to create/edit a Knowledge-Base. The run interface is illustrated below, and has the following functions:

- Pre run settings
- Target variable selection
- Question console
- How and Why console

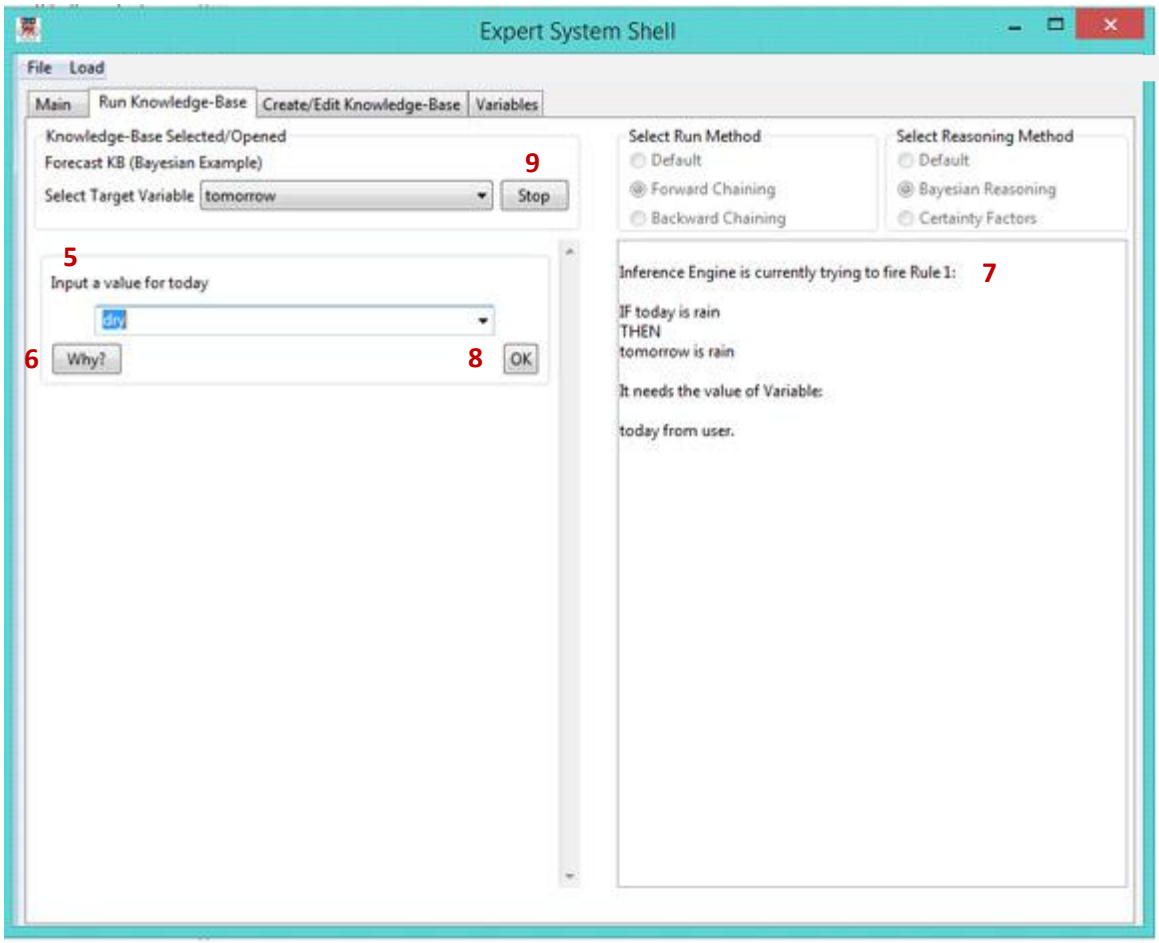


Once a Knowledge-Base is loaded, the user is required to choose how they wish to run the Knowledge-Base using the following process:

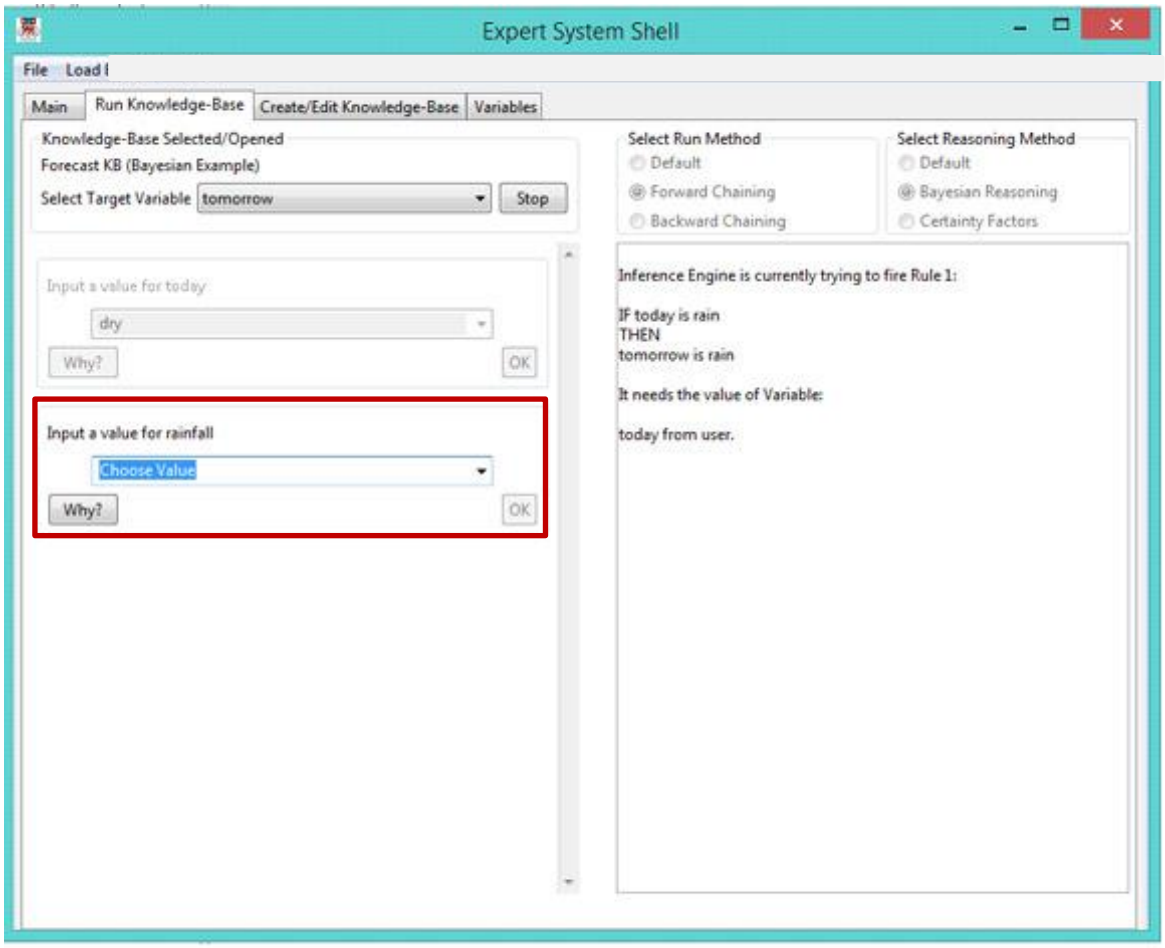


1. Select a run method from the Run Method Menu; Default, Forward Chaining, or Backward Chaining.
    a. Default – Runs forward chaining and is included for users who are new to Artificial Intelligence;
    b. Forward Chaining – the expert system will ask the user for all information necessary to determine a solution based on user input;
    c. Backward chaining- the user selects a solution in which the expert system will then ask the user for information to find evidence of that solution being correct.
2. Select a reasoning method from the Reasoning Method Menu; Default, Bayesian Reasoning, or Certainty Factor.
    a. Default – selected if there is no uncertainty in user input;
    b. Bayesian Reasoning- provides statistical information to give the user an indication of the chance that this solution is correct;
    c. Certainty Factor- is not based on statistical data and resembles the process the human mind takes in making a decision. This is the preferred method if statistical data is not available.
3. Then select a target variable/value from the drop down menu.

4. Select the run button. After the user has pressed the run button the following question box will appear in the question menu.

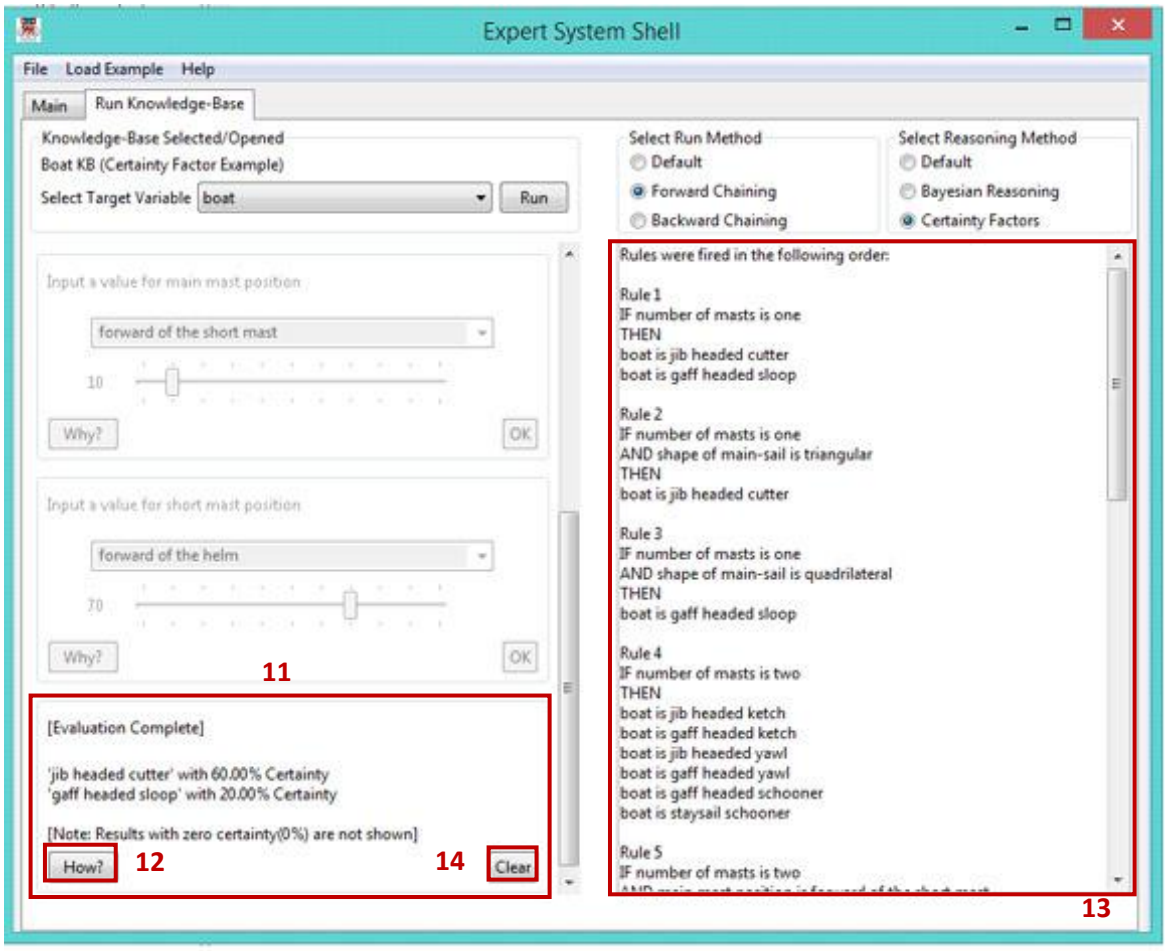


5. A question will appear in this box, to answer the question select a possible value from the drop down menu.
6. If the user wishes to know why a certain piece of information is required, press the why button.
7. The rule that is currently trying to be evaluated will appear in the box marked 7.
8. After the user has entered the value and is happy with why they needed to enter the information, press the ok button. Once the ok button is pressed, the user is unable to return and the question box will be disabled. A new question will appear under the first as shown below.
9. If the user wishes to stop the evaluation process, press the stop button. Once the stop button is pressed, it will turn into a run button. Pressing the stop button also clears the current evaluation process and thus the user cannot return to the half executed evaluation.

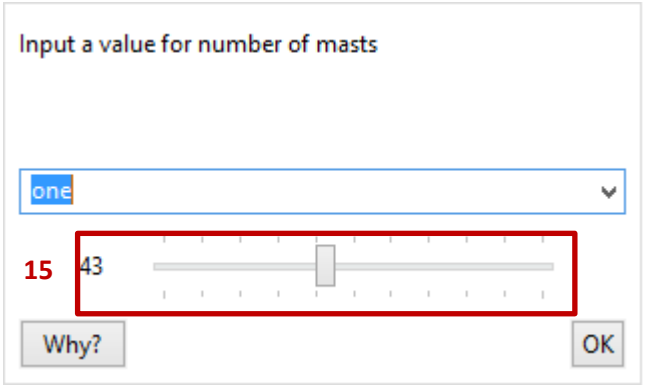


10

10. New questions will continue appear until the expert system has sufficient information to reach a conclusion. Once a solution is reached, the following window seen below will appear.
11. The evaluation window with the Bayesian coefficient displayed next to it.
12. If the user wishes to know how a decision was made, press the how button.
13. The rules that the program attempted to fire will be shown in this window.
14. Once the evaluation process is complete and the user is finished analysing the results, press the clear button clear the Knowledge-Base. Otherwise, press the run button to automatically clear and restart the evaluation process using the same set-up as the previous run.



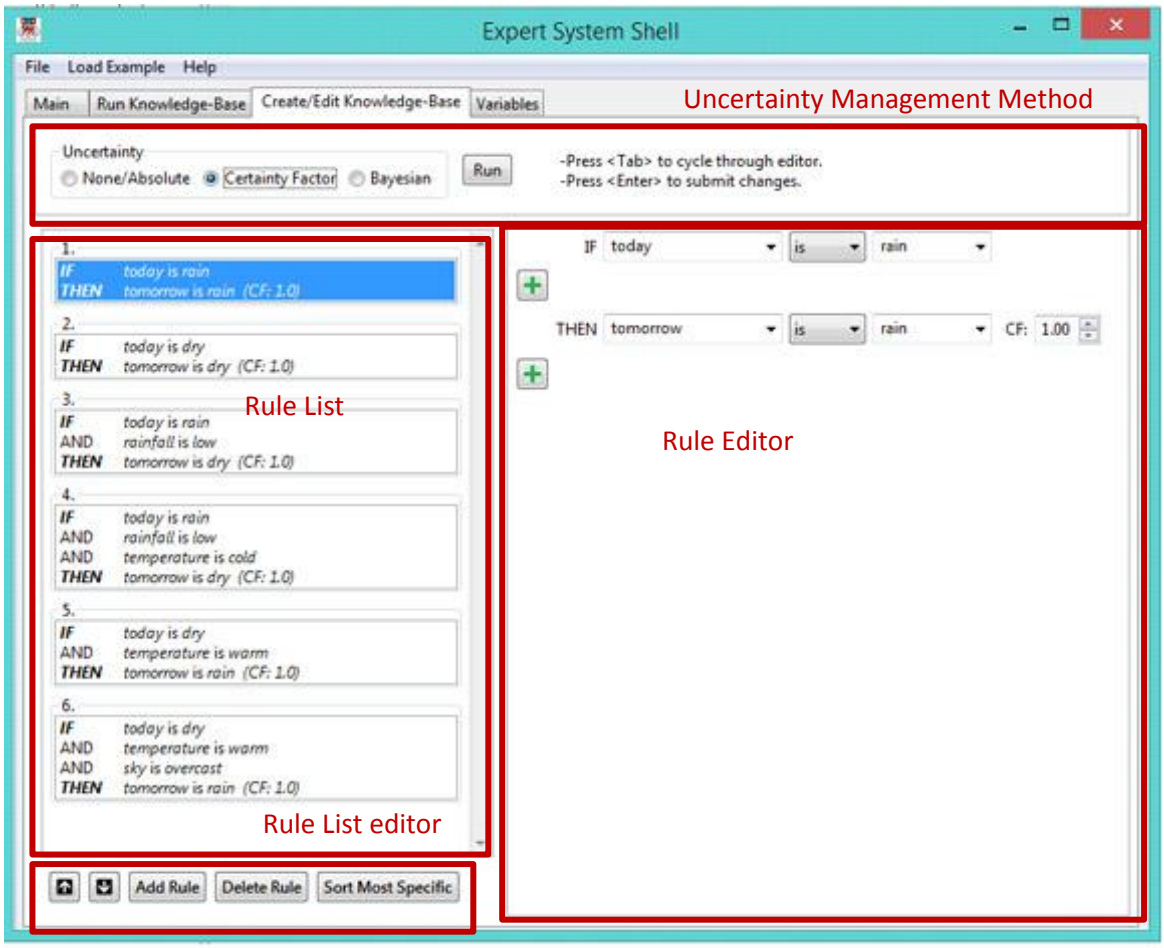
15. If the run has the uncertainty management method certainty factors selected the question box will also have a slider for you to choose your value of certainty see below



CREATE/EDIT KNOWLEDGE-BASE INTERFACE

The Create/Edit Knowledge-Base Interface has features which make it simple for the user to either create a new, or edit an existing, Knowledge-Base. The create/edit interface is illustrated below, and has the following functions:

- Uncertainty management method
- rule list
- rule editor
- rule list editor



The create/edit tab will look as above when you open a Knowledge-Base.

Uncertainty management method

There are three choices, simply select the one which is most relevant. If there is no uncertainty corresponding to the user's input and the possible solutions, select the 'None/Absolute' option. What the rule editor panel looks like will depend on which uncertainty method you have selected. The three options are:

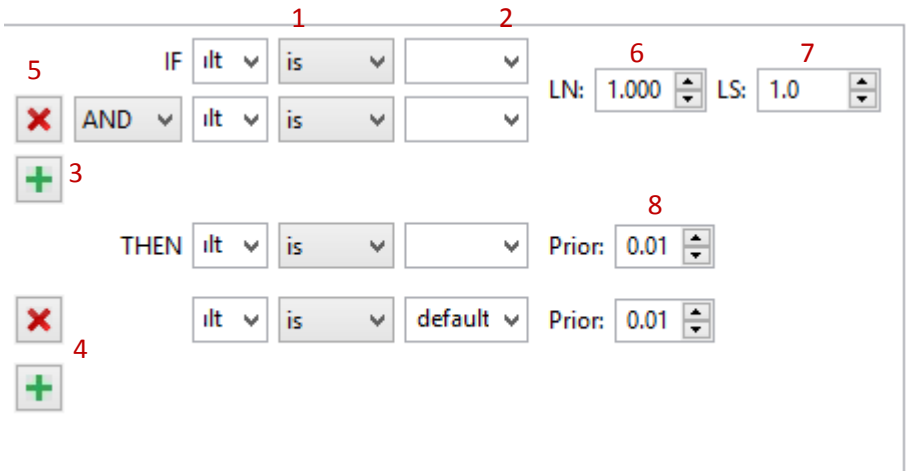
- None/Absolute
- Bayesian Reasoning
- Certainty Factor

Rule list

The rule list is the heart of the 'Create/Edit Knowledge-Base' Interface. The user will first need to select a rule from this list, at which point the rule editor will appear. If the user wishes to change the position of the rules you can use the rule list editor to do this.

Rule editor

Once the user has selected the rule they wish to edit, this panel will appear.



Each rule is always created with one antecedent and one consequent as each rule must have a least one of each. The following features are in the rule editor:

1. You will need to enter a variable into this field in which the rule will refer to. Once you have entered a variable it will automatically appear in the drop down list from then on.
2. This is the mathematical field operator and will give you the following options; =, !=, >, <, >=, <=, is and is not.
    a. Equals, =, to use this operator you have to have a numeric variable selected. The program will check if the variable is equal and will return true if it is. If the variable is not equal then the return will be false and the rule won't fire.
    b. Not equal, !=, this operator can also only be used on numeric variables. The program will check if the variable is not equal to the value entered and will return true or false as appropriate.
    c. Greater than, >, will operate only if a numeric variable is selected and will check if the value for that variable is greater than the Knowledge-Base and return true or false as appropriate.
    d. Less than, <, works the same as greater than except will check if the value is less than the one set.
    e. Greater than or equal to, >=, works the same as greater than but will also return true if the values are equal.
    f. Less than or equal to, <=, works the same as less than except will also return true if the values are equal.

g. Is, can only be used on linguistic variables and will check is the values are exactly the same this function is also case sensitive.
h. Is not, can also only be used on a linguistic variable but will return true if the values are not equal.

3. To add new antecedents to the rule click this button.
4. To add new consequents to the rule click this button.
5. To delete an antecedent or a consequent click the cross button next to the antecedent or consequent you need to delete.
6. The following field is only used if you wish to be able to run the uncertainty management method bayesian reasoning, this field is for likelihood of necessity.
7. The following field is only used if you wish to be able to run the uncertainty management method bayesian reasoning, this field is for likelihood of sufficiency.
8. The following field is only used if you wish to be able to run the uncertainty management method bayesian reasoning, this field is for the prior probability.
9. If you wish to use the certainty factor method this certainty factor field will appear see below.

Rule List Editor

This editor will only change the position of the rules it has the following functions

- Add new rule
- Delete rule
- Move rule up
- Move rule down
- sort

These functions are fairly self-explanatory it should be noted that if the sort/most specific button is pressed you will not be able to get your original rule list back again.

Example

For example we are going to create a new knowledge base called Forecast and add the following rules;

Rule 1

If today is rain

Then tomorrow is rain {cf 0.5}

Rule 2

If today is dry

Then tomorrow is dry {cf 0.5}

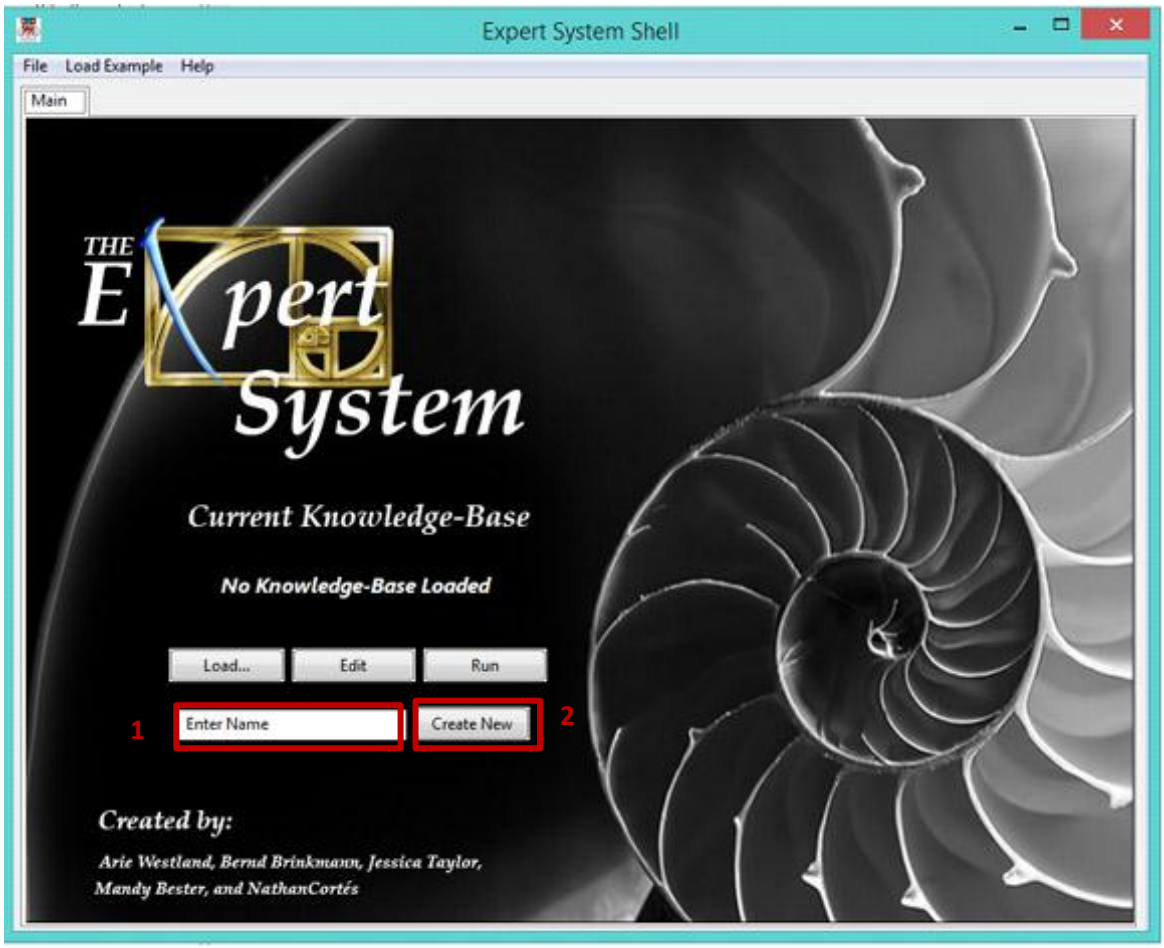
Rule 3

If today is rain

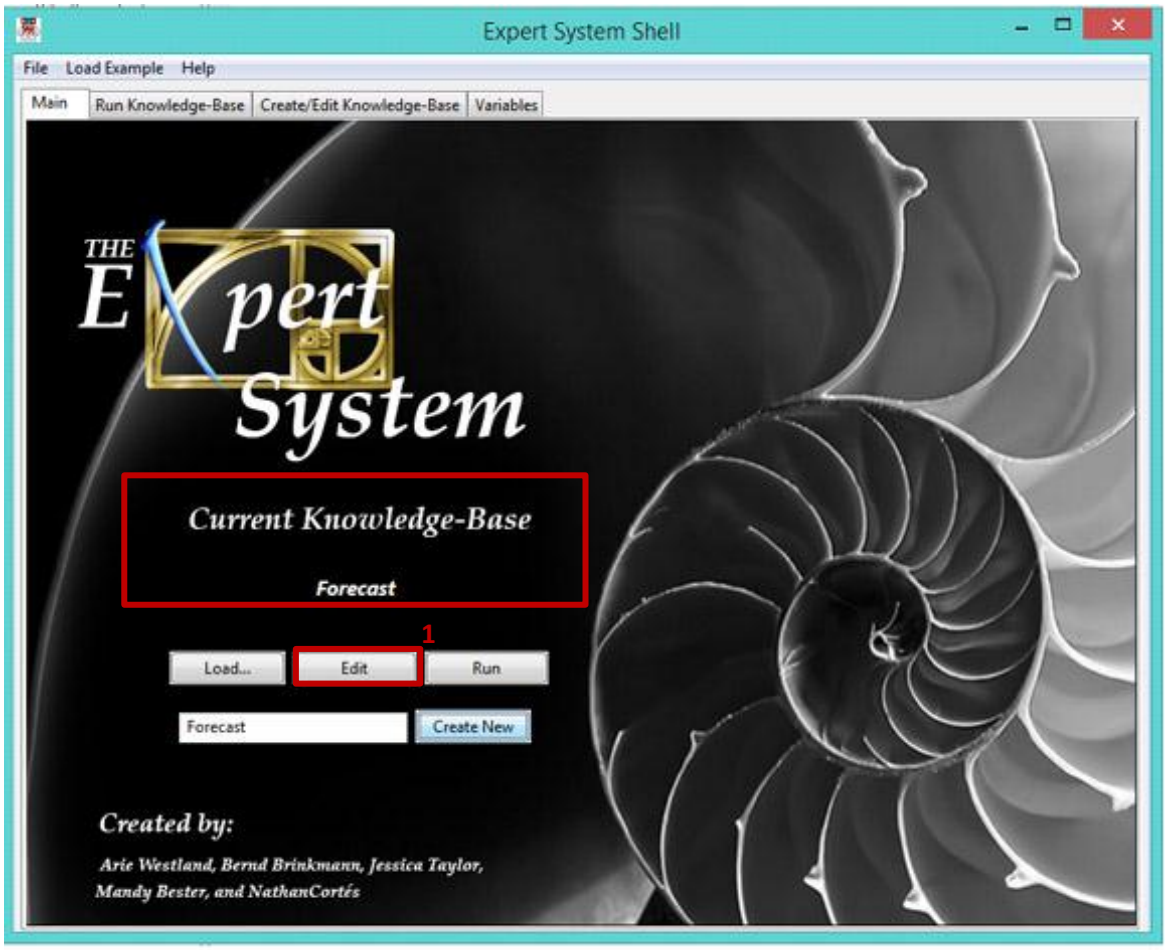
And rainfall is low

Then tomorrow is dry {cf 0.6}

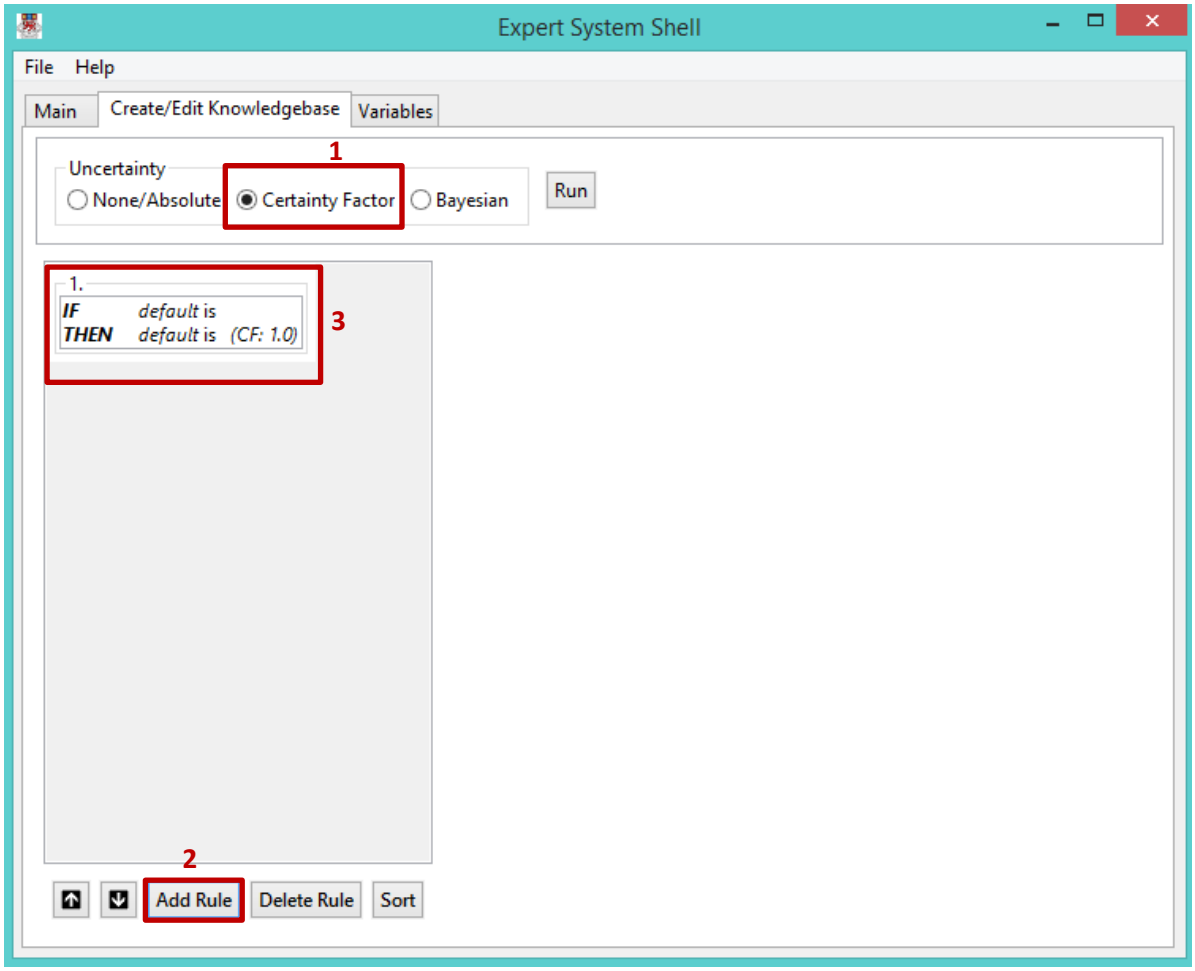
The first thing you will need to do is create the new knowledge base in the main screen. Type in “Forecast” and click the ‘Create New’ button.



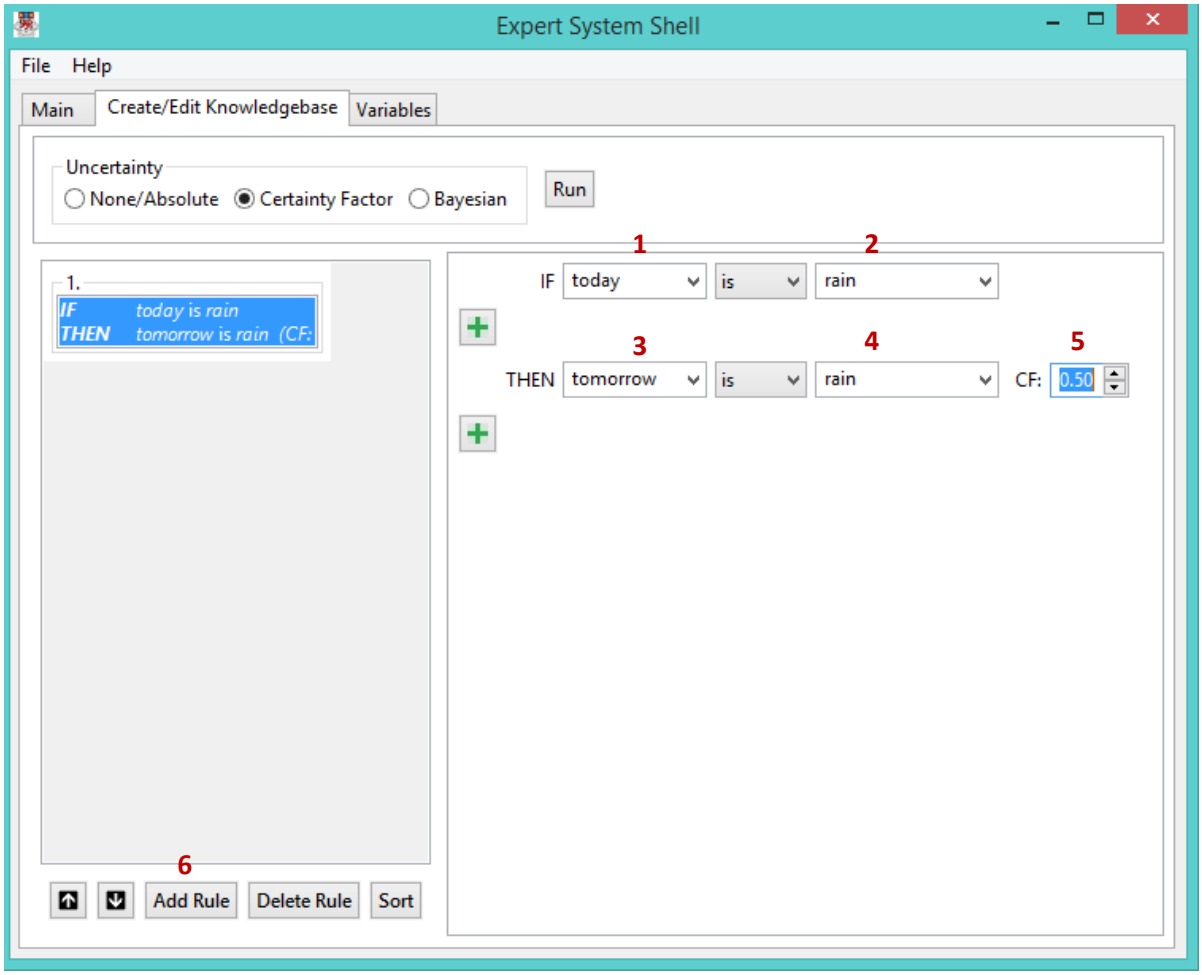
Now that you have done this you will see the word Forecast here. Then you will need to select the ‘Edit’ button



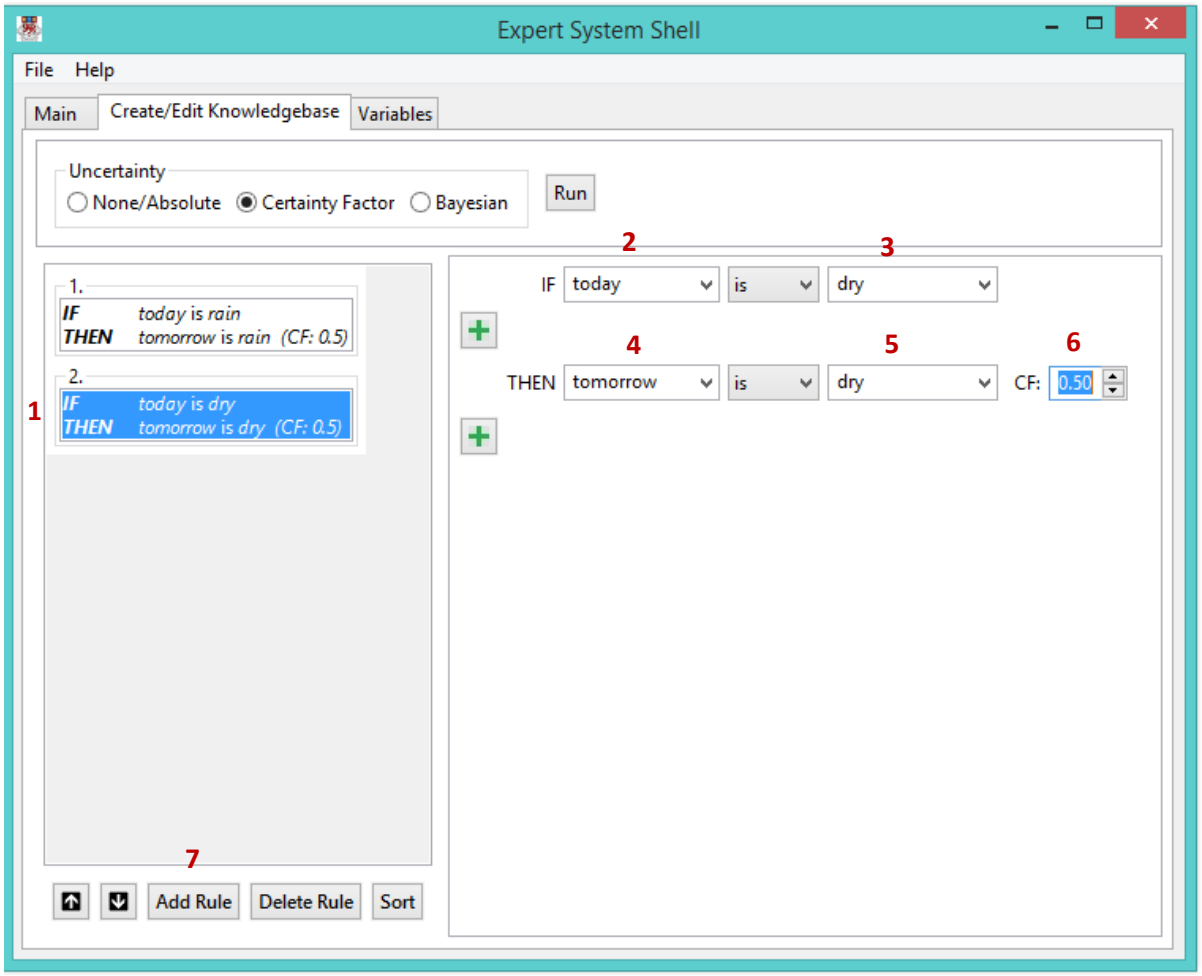
Now you will see the following tab open you will need to choose the type of uncertainty method you need to run and then click the add button to add a new rule. Select the rule that appears in the rule list to open the rule editor.



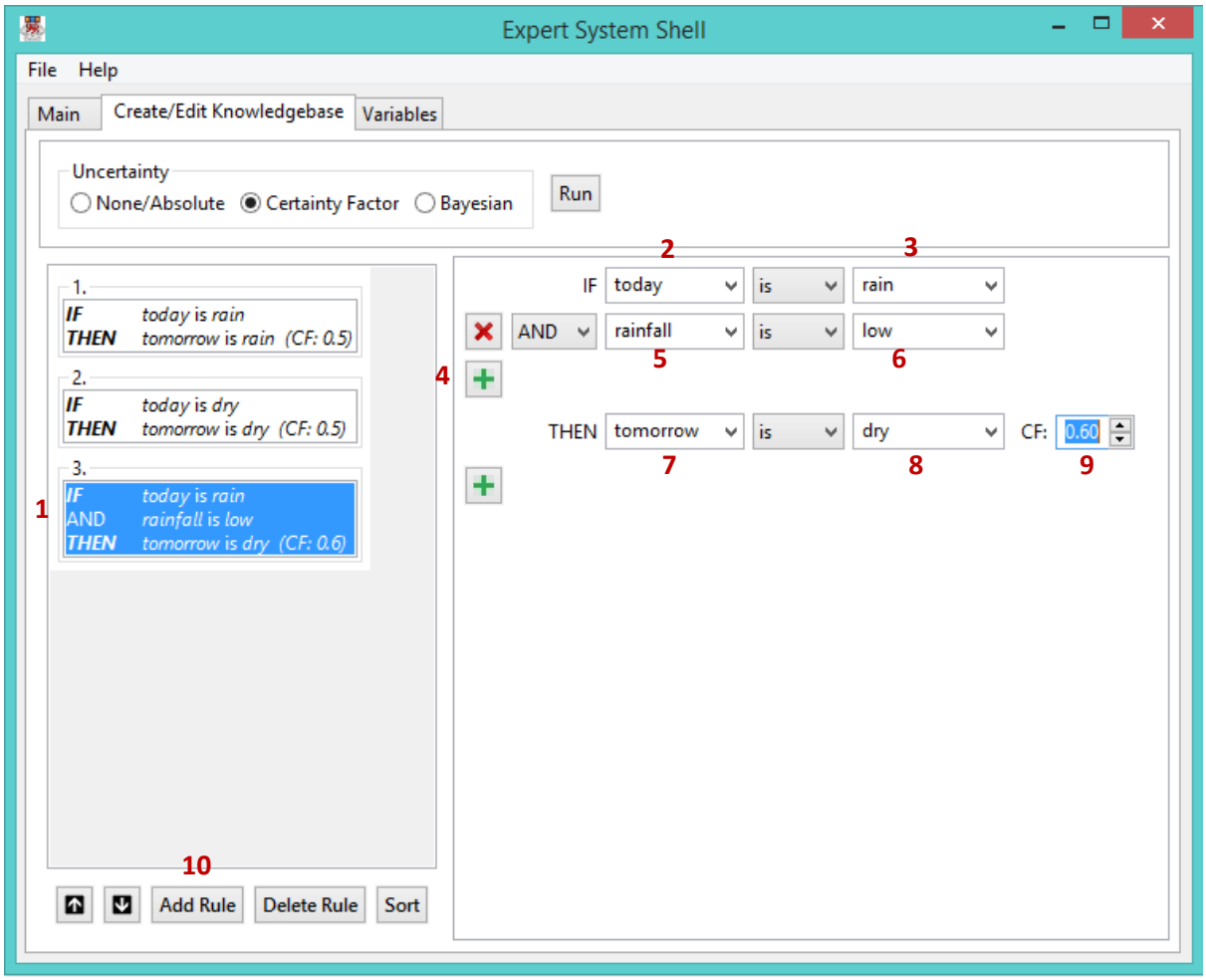
To enter the first rule add the word today to the first drop down box, the word rain to the 2nd, the word tomorrow to the 3rd and the word rain to the 4th. Then place the number 0.5 in the CF field and then select the add button again.



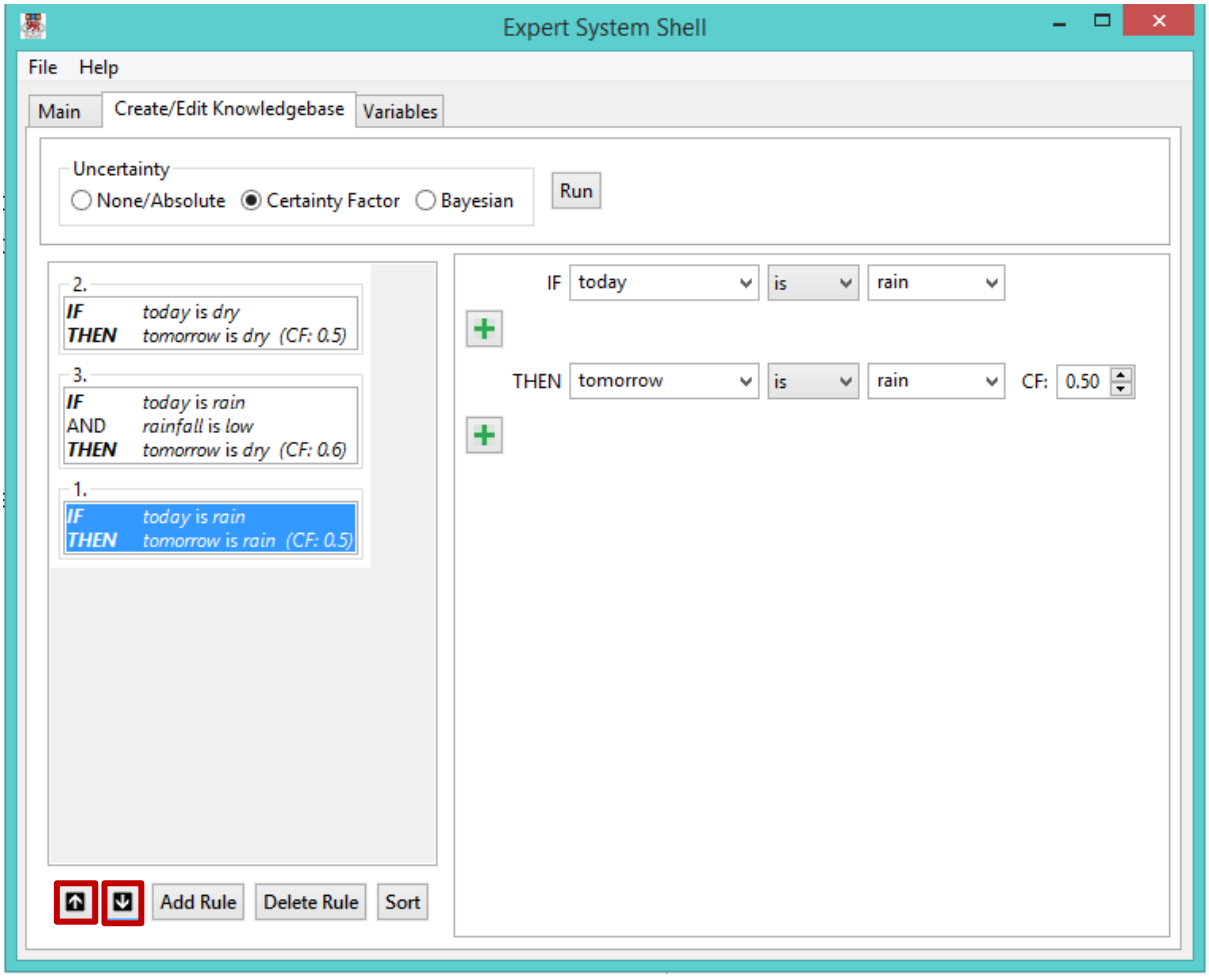
Select the new rule and add the word today to the first drop down menu, the word dry to the 2nd, the word tomorrow to the third and the word dry to the 4th. The add the number 0.5 to the CF field and press the add button again. Alternatively you could select the copy button in the previous step and just change the 2nd and 4th drop down boxes to dry.



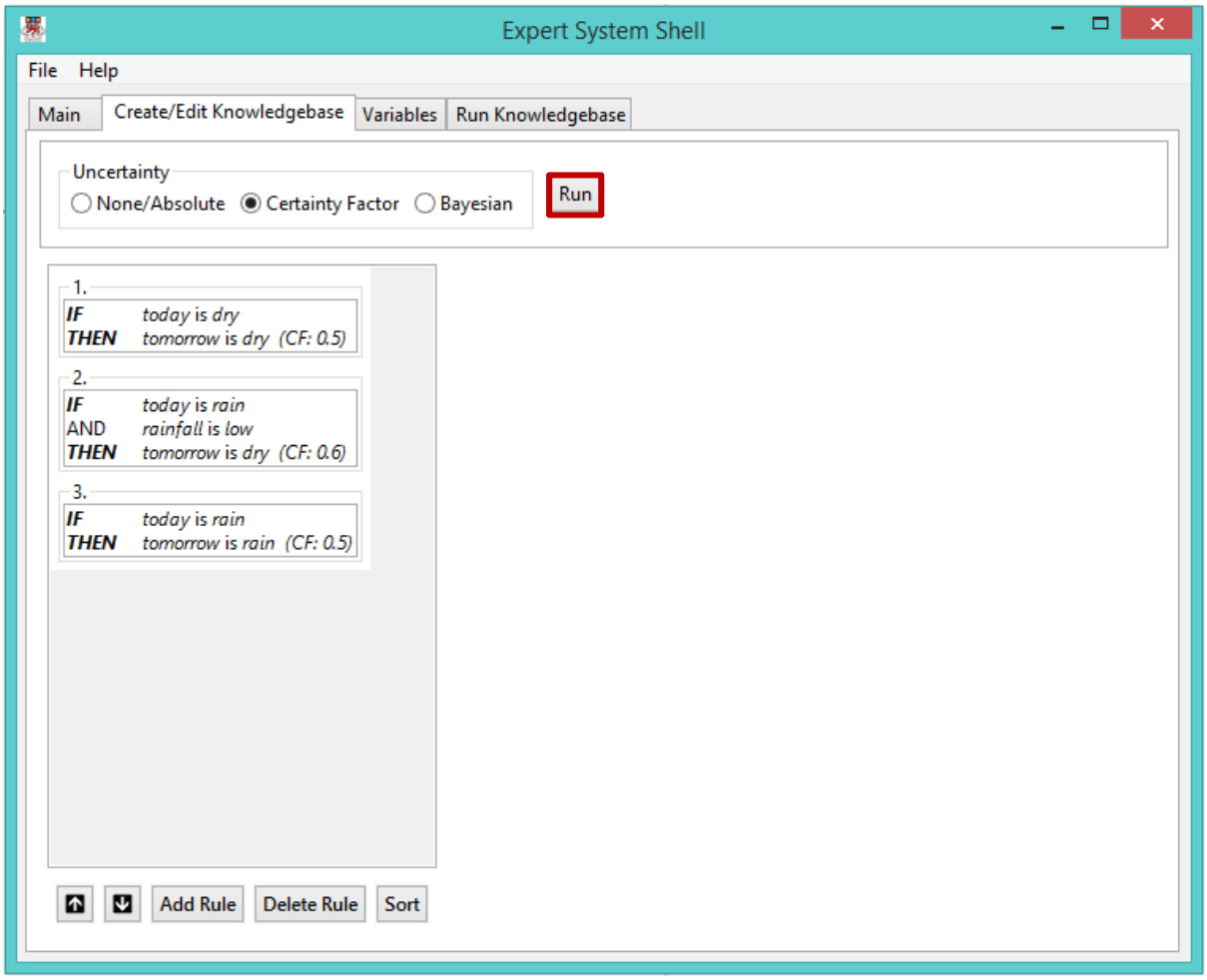
Enter the word today to the first drop down menu and rain to the second then press the plus button directly below. And enter the word rainfall to the new drop down menu and the word low to second new one. (Note: to add another consequent you simply press the plus button below the first consequent) Then you will need to add the word tomorrow to the first consequent drop down box and the word dry to the second and place the number 0.6 in the CF box.



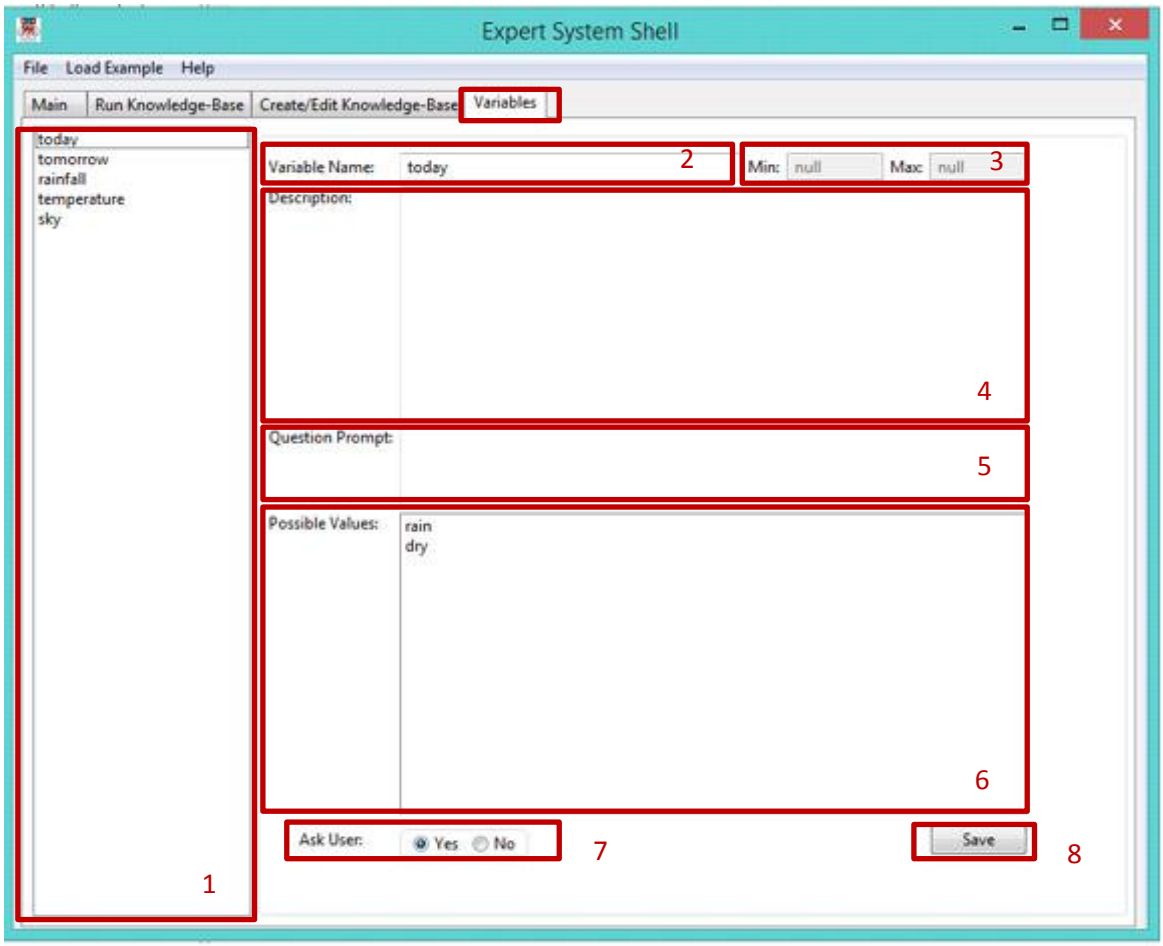
Now we want to switch the number of the rules do this by selecting the rule you wish to move for example rule 2 and press the up arrow. Then select the new second rule and click the down arrow in the rule editor. Your rule list should now look like this.



From here you will need to press run button which will redirect you to the run tab.



VARIABLES INTERFACE



1- Displays the list of variables in the current Knowledge-Base;
2- Edit the variable name ;
3- Enter a range of values for a numeric variable to take.
4- Add a description to the variable;
5- Question prompt - Ask a unique question, e.g. “How is the weather today?”. The Expert System Shell will ask the default question if nothing is entered here;
6- Displays the possible values the variable could take. These values will appear in a drop down menu when asking the user for the value of the variable in Run Knowledge-Base. The user cannot add to this list in this interface, and the value “unknown” will be automatically added to each variable in the drop down menu during an evaluation process;
7- Select whether or not the expert system requires user input for this variable, and thus whether a question needs to be asked;
8- Any changes made to this interface will not change the Knowledge-Base until save is pressed.

TECHNICAL DESIGN SPECIFICATIONS

This section describes the technical implementation of The Expert System Shell. The implementation of inference techniques – Forward Chaining and Backward Chaining; uncertainty techniques – Bayesian Reasoning and Certainty Factors; and conflict reasoning is discussed, with functionality of these run methods illustrated using flow charts for operation.

Inference Techniques

The inference engine links the rules provided by the Knowledge-Base with the facts in the database, and performs the reasoning in which the expert system reaches a solution. There are two principal methods that the inference engine employs to determine the order in which the rules of the expert system are fired, namely Forward Chaining and Backward Chaining.

Forward Chaining

Forward Chaining is a data-driven reasoning in which known data is used to fire as many rules as possible. This process is started by attempting to fire the top-most rule (hence when using Forward Chaining the order of rules becomes very important). Once that rule is completed, the expert system will continue down the rules list and fire any rule it can, obtaining any data it needs to along the way from the user. To demonstrate this, consider a simple traffic light example. The algorithm for this process is shown in the figure below.

Example: Forward Chaining-Traffic light

Consider the three rules below:

Rule 1

IF Traffic light IS RED

THEN action IS STOP

Rule 2

IF Traffic light IS GREEN

THEN action IS GO

Rule 3

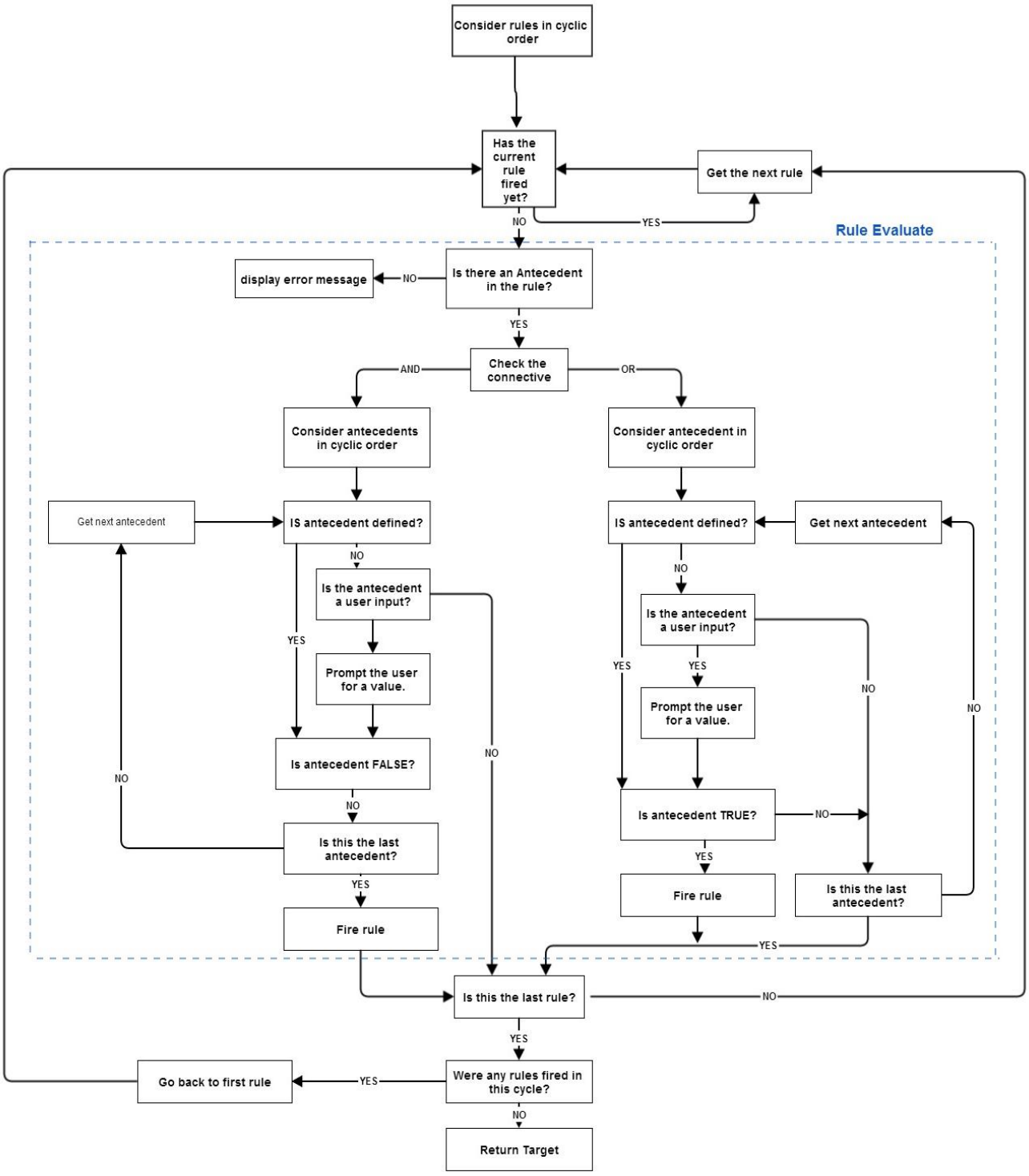
IF Traffic light IS RED

THEN action IS GO

It can be seen that there are two conflicting rules; rules 1 and 3. Assume that the traffic light is RED: When forward chaining is implemented, the expert system will fire rule 1 first and the action is

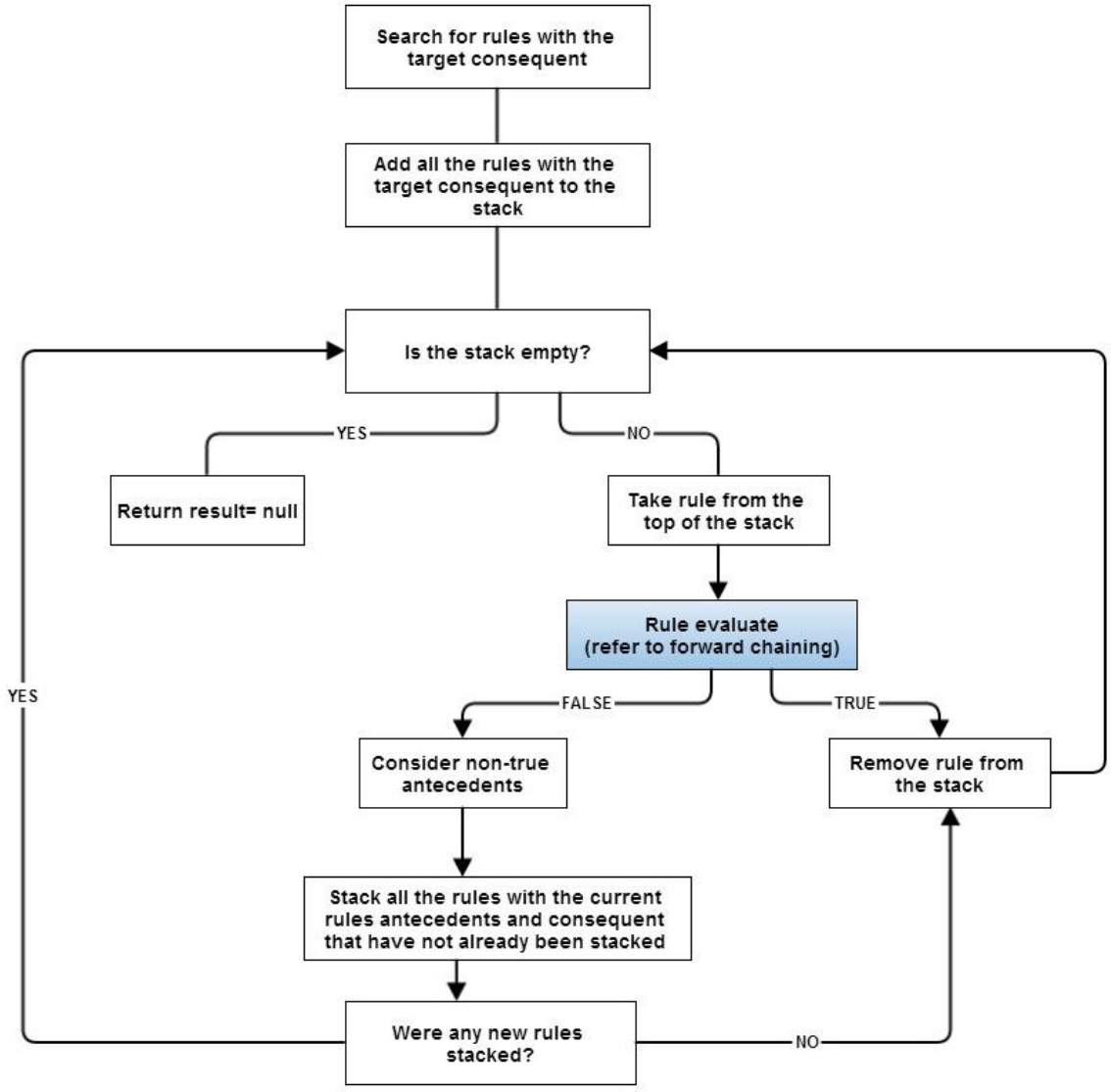
STOP. The system will then attempt to fire rule 2, which it cannot do, and then move on the rule 3. The action is then overwritten from before and is changed to GO. The final answer that the program will produce is therefore GO.

If you were to switch rules 1 and 3 and follow the same process the result would now be STOP. This is a clear example of the importance of the rule order when implementing forward chaining.



Backward Chaining

Backward Chaining is a goal-driven reasoning in which the expert system has a hypothetical solution that it attempts to prove, by finding evidence from rules within the Knowledge-Base. The first stage of this process searches the rule list for rules that contain the given consequent, and places these rules on a stack. The expert system then searches for the evidence to prove each of the stacked rules. Once the rule's evidence is completely known the process will be stopped. Figure 2 below shows the algorithm implemented. The rule evaluate block shown in blue is identical to the rule evaluate block represented by the blue dashed line in the figure above.



Example: Backward Chaining-Traffic light

Consider the same three rules presented in forward chaining.

Rule 1

IF Traffic light IS RED

THEN action IS STOP

Rule 2

IF Traffic light IS GREEN

THEN action IS GO

Rule 3

IF Traffic light IS RED

THEN action IS GO

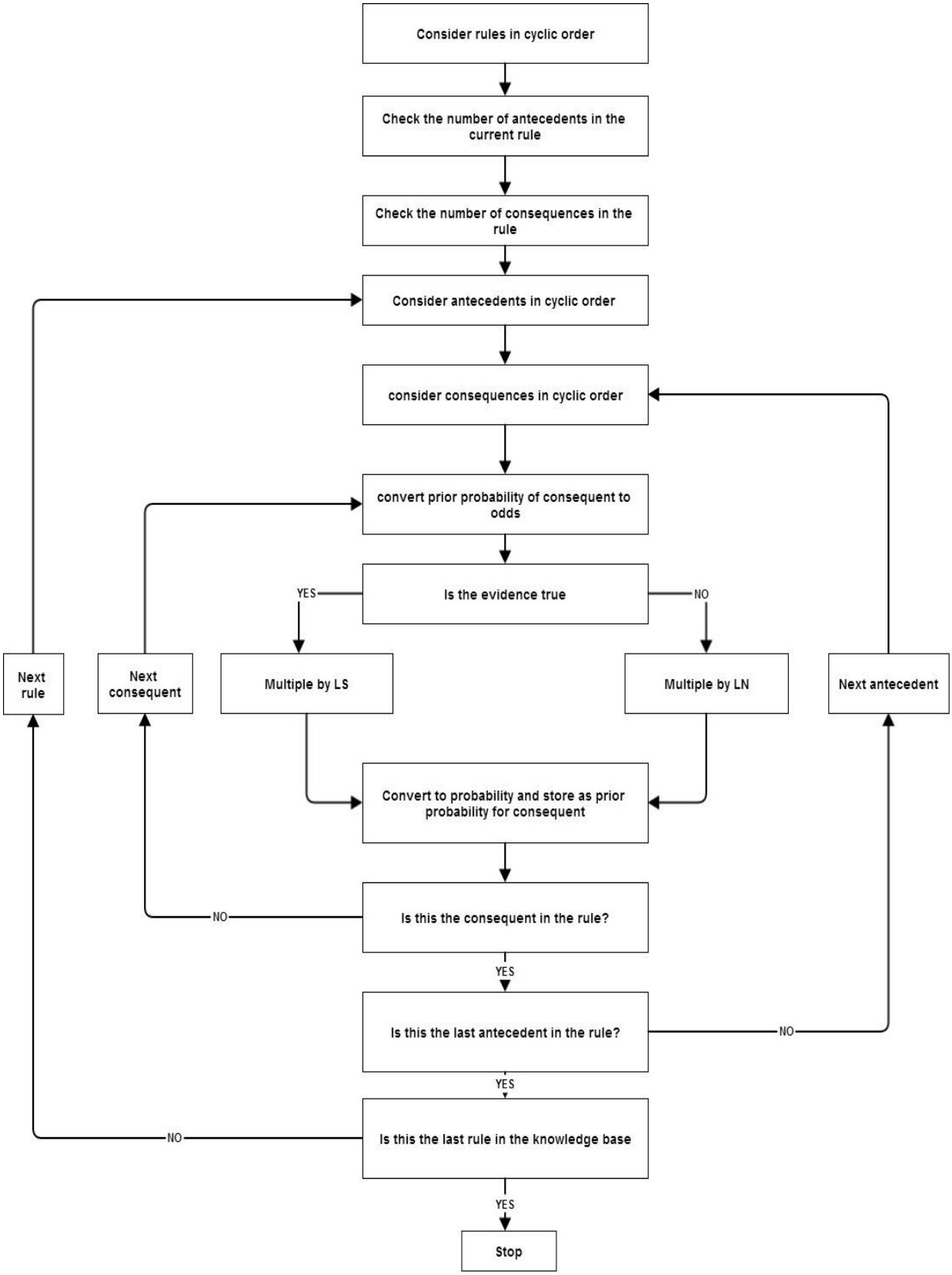
If the action the user wants to obtain is GO, the program will find all the rules with GO as a consequent, and asks the user a value for traffic light .If the user enters green, rule 2 will fire; if the user enters red, rule 3 will fire. Any rules with STOP as the consequent will be unable to fire.

Uncertainty Techniques

Expert systems are required to handle uncertainty, where exact knowledge that would enable the system to draw a perfectly reliable conclusion is incomplete, inconsistent or uncertain, and still draw valid conclusions. The most popular uncertainty management techniques, namely Bayesian reasoning and Certainty Factors, are implemented in The Expert System Shell and are discussed below.

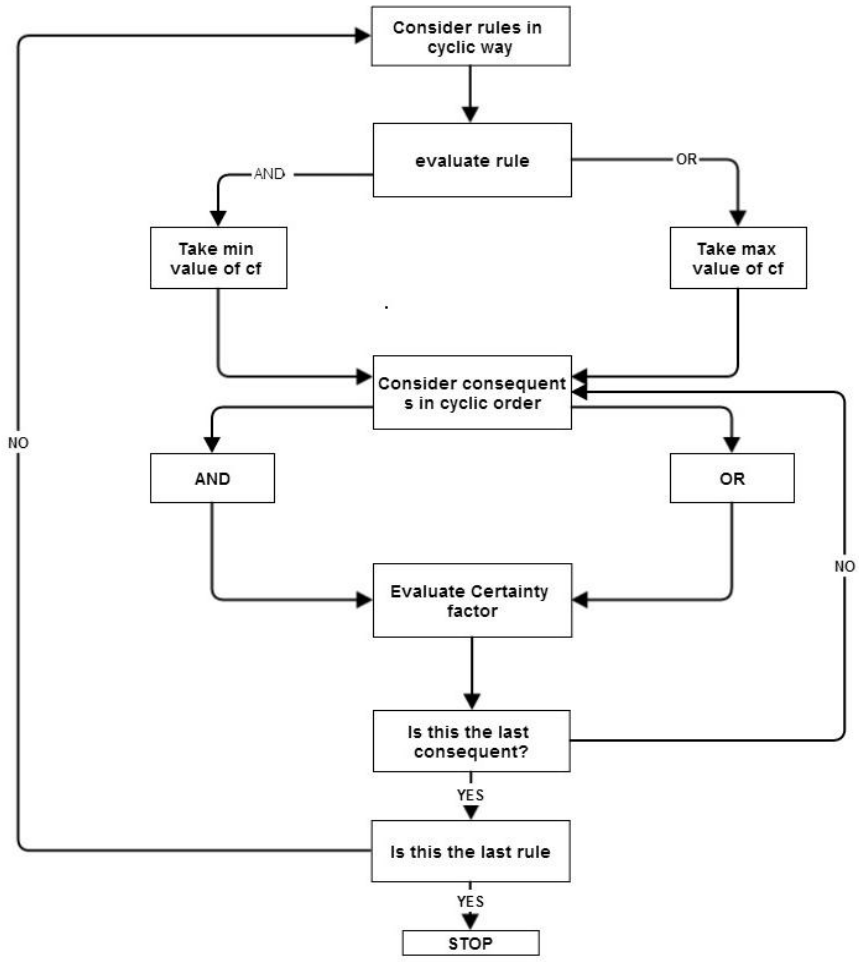
Bayesian reasoning

Bayesian Reasoning is one of the most well-known techniques for dealing with uncertainty in Expert Systems. The Bayesian Reasoning technique is based on Bayes theorem, which is the idea of conditional probability, which is the probability of the result being true based on the fact that the evidence is known to be true. This technique is good to use only if the appropriate statistical data is known or could be obtained easily. If statistical data is not obtainable, other methods should be used. The algorithm for the Bayesian reasoning is shown below.


Consider rules in cyclic order
Check the number of antecedents in the current rule
Check the number of consequences in the rule
Consider antecedents in cyclic order
consider consequences in cyclic order
convert prior probability of consequent to odds
Is the evidence true
YES
NO
Multiple by LS
Multiple by LN
Next rule
Next consequent
Next antecedent
Convert to probability and store as prior probability for consequent
Is this the consequent in the rule?
NO
YES
Is this the last antecedent in the rule?
NO
YES
Is this the last rule in the knowledge base
NO
YES
Stop

Certainty Factors

The Certainty Factors method is a popular alternative to Bayesian Reasoning as it does not require the same amount of statistical data. Certainty factors represent the expert's belief, and hence are a better representation of how a human would think. Certainty Factors are associated with both the antecedent and the consequent. They represent both the belief that the user has that their input is true, and the belief that the consequent has of being true based on the fact that the users input/evidence is true.



The evaluate certainty factor box the following equation is used.

$$\mathrm{cf}(\mathrm{cf1}, \mathrm{cf2}) = \mathrm{cf1} + \mathrm{cf2} \times (1 - \mathrm{cf1})$$

Where cf1 is the previous setting of the certainty factor for any rules fired with the same consequent, and cf2 is the certainty factor for the consequent of the current rule just fired.

Conflict Resolution

Conflict resolution is associated with selecting the appropriate rule to fire when multiple rules can be fired in a given cycle. I.e. antecedents of multiple rules can be satisfied simultaneously, although consequents may be very different. In this expert system, the “most specific rule” firing technique is employed. This technique is based on the assumption that a specific rule processes more information than a general rule. This is implemented into the expert system shell by rearranging the order of the rules when the “most specific first” option is selected in either the user or developer interfaces. The rules with the larger number of antecedents will be moved to the lower end of the knowledge base, while the rules with the lesser number of antecedents will be moved to the upper end of the knowledge base. Thus, in forward chaining, the rules with the larger number of antecedents will be fired last and will overwrite the conclusions drawn by the rules with a lesser number of antecedents; while for backward chaining, the expert system will attempt to prove the rule with the least number of antecedents first, before more complicated rules are attempted.

References

[1] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. 3rd edn, Addison and Wesley, Harlow, England, 2011