LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

John L. Darby

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico  87185 and Livermore, California  94550

Approved for public release; further dissemination unlimited.
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

SAND2007-1299
Unlimited Release
Printed March 2007

LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

John L. Darby
Security Systems Analysis
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0757

Abstract

LinguisticBelief is a Java computer code that evaluates combinations of linguistic variables using an approximate reasoning rule base. Each variable is comprised of fuzzy sets, and a rule base describes the reasoning on combinations of variables’ fuzzy sets. Uncertainty is considered and propagated through the rule base using the belief/plausibility measure. The mathematics of fuzzy sets, approximate reasoning, and belief/plausibility are complex. Without an automated tool, this complexity precludes their application to all but the simplest of problems. LinguisticBelief automates the use of these techniques, allowing complex problems to be evaluated easily. LinguisticBelief can be used free of charge on any Windows XP machine. This report documents the use and structure of the LinguisticBelief code, and the deployment package for installation client machines.
Acknowledgements

John C. Cummings of the Advanced Concepts Group at Sandia National Laboratories was the sponsor of this effort. John provided much direction, significant advice, and proposed a technique useful for interpreting results.

The author appreciates the careful review of this report by Greg Wyss and Peter Merkle of Sandia National Laboratories.
Contents

1 Introduction....................................................................................................................................11
  1.1 Overview of LinguisticBelief .................................................................................................12

2 Using LinguisticBelief..................................................................................................................22
  2.1 Structure of LinguisticBelief Graphical User Interface......................................................22
  2.2 Analysis Structure.............................................................................................................22
    2.2.1 Commonly Used Options .......................................................................................23
    2.2.2 Help Menu Options .............................................................................................24
    2.2.3 Utilities Menu .......................................................................................................27
  2.3 Creating an Analysis ..........................................................................................................29
    2.3.1 Adding Fuzzy Sets ...............................................................................................31
    2.3.2 Specify Input Linguistics ....................................................................................32
  2.4 Creating the Rule Base .......................................................................................................35
  2.5 Add Focal Elements .........................................................................................................36
  2.6 Calculate Results .............................................................................................................38
  2.7 Saving an Analysis ............................................................................................................47
  2.8 Changing an Existing Analysis .......................................................................................47
  2.9 Creating Another Specific Situation .................................................................................48

3 A Complex Example ..................................................................................................................49
  3.1 An Example Model for Evaluating Terrorist Scenarios ..................................................49
  3.2 Ranking Scenarios ............................................................................................................54

4 Special Cases ...........................................................................................................................59
  4.1 Probability .........................................................................................................................60
  4.2 Certainty ...........................................................................................................................62
  4.3 Total Ignorance ...............................................................................................................63

5 Summary....................................................................................................................................64

Appendix A. Code Architecture and Deployment .........................................................................65
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Figures

Figure 1-1. Belief/Plausibility as Bounds on Probability .............................................................12
Figure 1-2. Linguistics for Reasoning on “Quality of Life” ..........................................................13
Figure 1-3. FuzzySets for “Health” ..............................................................................................14
Figure 1-4. Approximate Reasoning Rule Base for “Quality of Life” ...........................................14
Figure 1-5. Focal Elements for “Health” of “John” .......................................................................15
Figure 1-6. Belief/Plausibility for “Wealth” for “John” ...............................................................16
Figure 1-7. Belief/Plausibility for “Quality of Life” for “John” ....................................................16
Figure 1-8. Model Expanded to include “Happiness” ....................................................................17
Figure 1-9. Approximate Reasoning Rule Base for “Happiness” ................................................18
Figure 1-10. Belief/Plausibility for “Happiness” for “John” ..........................................................18
Figure 1-11. Graphical Results for “Happiness” for “John” ...........................................................19
Figure 1-12. Belief/Plausibility Distribution for “Happiness” for “Janet” .......................................20
Figure 2-1. A New Analysis in LinguisticBelief ..........................................................................22
Figure 2-2. An Existing Analysis in LinguisticBelief ....................................................................23
Figure 2-3. Options under the File Menu .....................................................................................24
Figure 2-4. Options under the Help Menu ....................................................................................24
Figure 2-5. The Menu Item Help|About .......................................................................................25
Figure 2-6. The Menu Item Help|Analysis Limitations .................................................................25
Figure 2-7. The Menu Item Help|Changes to an Analysis .............................................................26
Figure 2-8. Options Under the Utilities Menu ...............................................................................27
Figure 2-9. The Menu Item Utilities|Help ....................................................................................27
Figure 2-10. Example of Warning with Utilities|DebugOn Selected ...........................................28
Figure 2-11. “Section 1 Example” with No Linguistic Variables ..................................................29
Figure 2-12. “Section 1 Example” After Adding Basic Linguistics ..............................................30
Figure 2-13. “Section 1 Example” After Adding Rule Linguistics ..............................................30
Figure 2-14. Options for the Basic Linguistic “Health” .................................................................31
Figure 2-15. Options for the Rule Linguistic “Health” .................................................................31
Figure 2-16. Fuzzy Sets for “Health” ............................................................................................32
Figure 2-17. Specifying Input Linguistics for the Rule Linguistic “Quality of Life” ......................32
Figure 2-18. Dialog for Adding Input Linguistics to “Quality of Life” .........................................33
Figure 2-19. “Quality of Life” with Input Linguistics Specified ...................................................33
Figure 2-20. “Happiness” with Input Linguistics Specified ............................................................34
Figure 2-21. Error Message if Circular Logic Is Detected ............................................................34
Figure 2-22. Empty Rule Base for “Quality of Life” .....................................................................35
Figure 2-23. Complete Rule Base for “Quality of Life” ...............................................................35
Figure 2-24. Complete Rule Base for “Happiness” .......................................................................36
Figure 2-25. Adding Focal Elements for “Health” for “John” .......................................................37
Figure 2-26. Warning of Incomplete Evidence for a Basic Linguistic ........................................37
Figure 2-27. All Focal Elements Added for “Health” for “John” ....................................................38
Figure 2-28. Error Message for Attempt to Add A Focal Element that Would Result in Sum of Evidence Over All Focal Elements Greater than One .......................................................38
Figure 2-29. Generating Results for “Quality of Life” for “John” ................................................39
Figure 2-30. Results for “Quality of Life” for “John”.................................................................39
Figure 2-31. “Quality of Life” for “John” with a Family Selected............................................40
Figure 2-32. Fuzzy Sets, Not Ordered ..........................................................................................40
Figure 2-33. Fuzzy Sets, Partially Ordered .................................................................................41
Figure 2-34. Fuzzy Sets, All Ordered .........................................................................................41
Figure 2-35. Graphical Results for “Quality of Life” for “John” ................................................42
Figure 2-36. “Happiness” for “John” .........................................................................................43
Figure 2-37. “Happiness” Fuzzy Sets Ordered “Best” to “Worst” ............................................43
Figure 2-38. Graphical Results for “Happiness” for “John” .....................................................44
Figure 2-39. “Wealth” for “John” .............................................................................................45
Figure 2-40. “Wealth” Fuzzy Sets Ordered “Best” to “Worst” ...............................................45
Figure 2-41. Graphical Results for “Wealth” for “John” ..........................................................46
Figure 2-42. Dialog for Changing Fuzzy Sets and Input Linguistics for a RuleLinguistic ......47
Figure 2-43. Dialog for Changing Fuzzy Sets and Focal Elements for a BasicLinguistic .......47
Figure 2-44. Dialog for Removing all Focal Elements ...............................................................48
Figure 3-1. Analysis Model for Risk of a Scenario .................................................................49
Figure 3-2. Rules for Risk .......................................................................................................50
Figure 3-3. Results for Risk for Scenario 1 .............................................................................51
Figure 3-4. Graphical Results for Risk for Scenario 1 ...............................................................52
Figure 3-5. Results for Risk for Scenario 2 .............................................................................53
Figure 3-6. Graphical Results for Risk for Scenario 2 ...............................................................54
Figure 3-7. Graphical Presentation for Ranking of Scenario 1 ................................................56
Figure 3-8. Graphical Presentation for Ranking of Scenario 2 ................................................56
Figure 3-9. Graphical Presentation for Ranking of Happiness of John ......................................57
Figure 3-10. Graphical Presentation for Ranking of Happiness of Janet ..................................58
Figure 4-1. Graphical Results for “Health” for “John” ............................................................59
Figure 4-2. Graphical Results for “Health” for “Fred” ............................................................61
Figure 4-3. Graphical Results for “Health” for “Richard” .......................................................62
Figure 4-4. Graphical Results for “Health” for “Mary” ...........................................................63

Listings

Listing 2-1. Example of Information Provided when Removing a Linguistic Variable with Utilities|DebugOn Selected .............................................................................................................28
Listing 3-1. Results from RankScenarios for Two Terrorist Scenarios ..................................55
Listing 3-2. Results from RankScenarios for Happiness .......................................................57
Acronyms

gui                      graphical user interface
IDE                      Integrated Development Environment
LDRD                     Laboratory Directed Research and Development
SNL                      Sandia National Laboratories
xml                      extended markup language
Executive Summary

This report documents the LinguisticBelief Java computer code, version 1.0. LinguisticBelief is a computer tool for the evaluation of combinations of linguistic variables. The Laboratory Directed Research and Development (LDRD) program at Sandia National Laboratories (SNL) sponsored this work. The work was performed between October 1, 2006, and January 23, 2007, at SNL in Albuquerque, New Mexico.

Using the code, linguistic variables are combined by approximate reasoning on fuzzy sets for the variables. The belief/plausibility measure of uncertainty is used to capture and propagate uncertainty for the linguistic variables. The tool is appropriate for the evaluation of variables that are difficult to express numerically. The belief/plausibility measure of uncertainty allows evaluation where there is significant epistemic uncertainty.

The mathematics of fuzzy sets, approximate reasoning, and belief/plausibility are complex. Without an automated tool, this complexity precludes their application to all but the simplest of problems. LinguisticBelief was developed to automate the use of these techniques, allowing complex problems to be easily evaluated. The code has a graphical user interface (gui) that can be used to easily create custom reasoning rule bases, apply degrees of evidence, and evaluate the rule base considering uncertainty.

Using a simple example, this report summarizes the techniques of approximate reasoning, fuzzy sets, and the belief/plausibility measure of uncertainty as implemented in LinguisticBelief. Also, the process by which a user of the tool can create a customized reasoning model is explained.

The LinguisticBelief code is being used to evaluate and rank-order potential terrorist attack scenarios. To show the use of the code to a real-world problem, this report provides summary, unclassified information on how the code is used for that application.

The belief/plausibility measure of uncertainty includes the probability measure of uncertainty as a special case, which is discussed in this report. Also, the special cases of “no uncertainty” and “total ignorance” are discussed.

An appendix summarizes the architecture and deployment of the code.

The compiled LinguisticBelief code—all with the Java runtime environment, required libraries, and a batch file for executing the code in Windows XP—all have been packaged together for deployment. The deployed package can be installed and used at no cost.

In summary, the LinguisticBelief code is a fully functional tool for creating and evaluating complex combinations of linguistic variables, including uncertainty. This report documents the use of the code.
1 Introduction

LinguisticBelief is a computer tool for the evaluation of combinations of linguistic variables. Using the tool, linguistic variables are combined by approximate reasoning on fuzzy sets for the variables. The belief/plausibility measure of uncertainty is used to capture and propagate uncertainty for the linguistic variables. The tool is appropriate for the evaluation of variables that are difficult to express numerically. The belief/plausibility measure of uncertainty allows evaluation where there is significant epistemic uncertainty.

The mathematics of fuzzy sets, approximate reasoning, and belief/plausibility are complex. Without an automated tool, this complexity precludes their application to all but the simplest of problems. LinguisticBelief automates the use of these techniques, allowing complex problems to be easily evaluated. The code has a graphical user interface (gui) that can be used to easily create custom reasoning rule bases, apply degrees of evidence, and evaluate the rule base considering uncertainty.

LinguisticBelief is a 100% Java application written by the author. It was developed using the free NetBeans 5 Integrated Development Environment (IDE) from Sun Microsystems. LinguisticBelief requires “Java 5” (JSDK5) or higher as it uses features new to the language, such as generics.

LinguisticBelief uses the JFreeChart class library to create graphs of results. [Gilbert]

The compiled LinguisticBelief code–along with the Java runtime environment, required libraries, and a batch file for executing the code in Windows XP–all have been packaged together for deployment. The deployed package can be installed and used at no cost, and can be obtained by contacting the author of this report at jldarby@sandia.gov.

The techniques used in LinguisticBelief are documented in detail in a reference that summarizes the belief/plausibility measure of uncertainty, fuzzy sets, and approximate reasoning as applied to the linguistic evaluation approach used in LinguisticBelief. [Darby]

For the purposes of this report, the following summary information is provided. When we use linguistics (words) to classify events, the words have a type of uncertainty called “vagueness.” For example, yesterday was “sunny,” public confidence in the stock market is “high,” etc. Vagueness is uncertainty as to how to classify a known event. For example, assume we know how tall John is, but instead of saying John is 6 feet 2 inches tall we categorize John as “tall” without a precise definition of “tall.” The linguistic (word) “tall” is vague. Vagueness can be addressed using the mathematics of fuzzy sets.

The belief/plausibility measure of uncertainty from the Dempster/Shafer Theory of Evidence is an extension of the probability measure of uncertainty that can better capture epistemic uncertainty. Belief/plausibility is a superset of probability and under certain conditions, belief and plausibility both become probability. Under other conditions, belief/plausibility become necessity/possibility, respectively. Belief/plausibility addresses a type of uncertainty called “ambiguity.” The uncertainty associated with predicting an event in the future is ambiguity.
A simple example illustrates the difference between aleatory (stochastic or “random”) uncertainty and epistemic (state-of-knowledge) uncertainty, and the use of a belief/plausibility measure. Consider a fair coin, heads on one side, tails on the other, with each side equally likely. The uncertainty as to the outcome of a toss—heads or tails—is aleatory. The probability of heads is ½ and the probability of tails is ½. The uncertainty is due to the randomness of the toss. Suppose, however, that I do not know the coin is fair; the coin could be biased to come up heads, or the coin could even be two-tailed. Now that I have epistemic uncertainty, my state of knowledge is insufficient to assign a probability to heads or tails; all I can say is the likelihood of heads (or tails) is somewhere between 0 and 1. To consider epistemic uncertainty as well as aleatory uncertainty, belief/plausibility can be used as the measure of uncertainty. With total ignorance about the coin, the belief that the toss will be heads is 0 and the plausibility that the toss will be heads is 1; similarly, the belief that the toss will be tails is 0 and the plausibility that the toss will be tails is 1. Belief/plausibility form an interval that can be interpreted as giving the lower and upper bound of probability. If I have enough information, both belief and plausibility reduce to a single value, probability. Figure 1-1 illustrates this concept. Epistemic uncertainty can be reduced with more information. If I toss the coin a few times and a heads and a tails occur, I know the coin is two-sided; with more tosses I can evaluate the fairness of the coin. Aleatory uncertainty cannot be reduced with more information.

![Belief/Plausibility as Bounds on Probability](image)

Approximate reasoning is a rule base for combining fuzzy sets for different linguistic variables. For example, “Bad” “Health” and “Poor” “Wealth” cause “Not so Good” “Quality of Life.”

The LinguisticBelief code uses an approximate reasoning rule base to evaluate combinations of linguistic variables expressed as fuzzy sets, considering uncertainty using the belief/plausibility measure.

### 1.1 Overview of LinguisticBelief

An overview of LinguisticBelief is summarized in this section using a simple example. Section 2.3 shows how this example is created in LinguisticBelief.

Assume we wish to reason on “Quality of Life” based on “Health” and “Wealth.” “Health” and “Wealth” are basic linguistic variables, and “Quality of Life” is a rule-based linguistic variable that it is evaluated by a set of rules for combining “Health” and “Wealth.” Figure 1-2 shows these linguistics entered into LinguisticBelief.
For “Health” we chose to use the fuzzy sets “Bad,” “Moderate,” and “Excellent.” For “Wealth” we use the fuzzy sets “Poor,” “Middle Class,” and “Rich.” For “Quality of Life” we use the fuzzy sets “Good” and “Not so Good.” Figure 1-3 shows the fuzzy sets for “Health” entered into LinguisticBelief.

Figure 1-2. Linguistics for Reasoning on “Quality of Life”
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Figure 1-3. Fuzzy Sets for "Health"

Figure 1-4 shows the user-created approximate reasoning rule base used to evaluate "Quality of Life" based on "Health" and "Wealth." The rule base specifies the mapping of combinations of input fuzzy sets to output fuzzy sets. For example, the rule base in Figure 1-4 indicates that "Bad" "Health" and "Poor" "Wealth" result in "Not so Good" "Quality of Life."

Figure 1-4. Approximate Reasoning Rule Base for "Quality of Life"

---

1 The variables are assumed to be non-interacting. [Darby].
This model can be used to reason on Quality of Life for any individual.

The Quality of Life for a specific individual is evaluated by assigning degrees of evidence to collections of fuzzy sets for each basic linguistic variable. Assume we have the following focal elements (evidence over families of fuzzy sets) for the “Health” of a specific individual named “John”:

- 0.8 to {“Bad,” “Moderate”}, and
- 0.2 to {“Moderate,” “Excellent”}.2

Assume we have the following focal elements for the “Wealth” of “John”:

- 0.3 to {“Middle Class”}, and
- 0.7 to {“Poor,” “Middle Class”}.

Figure 1-5 shows the focal elements for the “Health” of “John” entered into LinguisticBelief; similarly, the focal elements for the “Wealth” of “John” are entered into the code.

After focal elements for all basic linguistic variables are entered, the belief/plausibility for any linguistic variable (basic or rule-based) can be calculated. Figure 1-6 shows the belief/plausibility for “Wealth” for “John,” and Figure 1-7 shows the belief/plausibility for “Quality of Life” for “John.”

---

2 As discussed in a reference, the fuzziness of the sets is considered in the assignment of degrees of evidence. [Darby] For linguistic convolution in the rule base, the sets are treated as crisp. Both belief and plausibility are probability if all the focal elements are singletons; see Section 4.1 of this report.
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

**Figure 1-6. Belief/Plausibility for “Wealth” for “John”**

**Figure 1-7. Belief/Plausibility for “Quality of Life” for “John”**
The belief/plausibility results from the code provide three categories of information for the specific linguistic variable of interest. First, the [belief, plausibility] interval for each of the fuzzy sets is given; for example, Figure 1-6 shows the following for “Wealth” for “John”:

**BELIEF AND PLAUSIBILITY FOR FUZZY SETS:**
- Poor has [Belief, Plausibility] Interval of: [0.000, 0.700]
- Middle Class has [Belief, Plausibility] Interval of: [0.300, 1.000]
- Rich has [Belief, Plausibility] Interval of: [0.000, 0.000]

Second, the focal elements for the linguistic variable are provided; for example, Figure 1-6 shows the following focal elements for “Wealth” for “John”:

**FOCAL ELEMENTS:**
- Middle Class. Evidence is 3.000e-01.
- Poor & Middle Class. Evidence is 7.000e-01.

Third, the [belief, plausibility] interval for any family of fuzzy sets can be calculated; for example, Figure 1-6 shows the [belief, plausibility] for the family

{“Middle Class,” “Rich”} for the “Wealth” for “John” as:

[3.000e-01, 1.000e+00].

This information answers such questions as: “What is the belief/plausibility that “John” is wealthier than ‘Poor’?” (This is the belief/plausibility for {“Middle Class,” “Rich”} which is 0.3/1.0.)

Assume we expand the model to reason on “Happiness,” another rule-based linguistic variable, as a combination of “Quality of Life” and a new basic linguistic variable “Outlook on Life.” The fuzzy sets for “Outlook on Life” are specified as: “Pessimist” and “Optimist.” The fuzzy sets for “Happiness” are chosen to be: “Depressed,” “Accepting,” and “Very Happy.” Figure 1-8 shows the expanded model in LinguisticBelief, and Figure 1-9 shows the user-specified approximate reasoning rule base for “Happiness” based on “Quality of Life” and “Outlook on Life.”

![LinguisticBelief Application](image)

**Figure 1-8. Model Expanded to include “Happiness”**
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Figure 1-9. Approximate Reasoning Rule Base for “Happiness”

Note that “Happiness,” a rule-based linguistic, is a combination of “Quality of Life,” another rule-based linguistic, and “Outlook on Life,” a basic linguistic. The LinguisticBelief code allows a rule-based linguistic to be a combination of linguistics, basic or rule-based. Circular logic is not allowed; that is, no rule-based linguistic can have itself as an input linguistic. The code recursively checks any proposed input linguistic that is itself a rule-based linguistic for circular logic.

Assume the following focal elements for “Outlook on Life” for “John:”

{“Pessimist”} with evidence 0.02, and
{“Pessimist,” “Optimist”} with evidence 0.98.

Figure 1-10 shows the belief/plausibility results for “Happiness” for “John.”

Figure 1-10. Belief/Plausibility for “Happiness” for “John”
Using the **Plot Results** button in Figure 1-10, the graphical results in Figure 1-11 are produced.

![Graphical Results for “Happiness” for “John”](image)

**Figure 1-11. Graphical Results for “Happiness” for “John”**

The top graph in Figure 1-11 is the “likelihood” of each fuzzy set, where likelihood is a belief/plausibility interval. The fuzzy sets for “Happiness” in Figure 1-11 are ordered from most happy to least happy as: “Very Happy,” “Accepting,” and “Depressed.” Using this ordering, the likelihood of exceedance for each of these fuzzy set can be calculated. The lower graph in Figure 1-11 shows a high belief/plausibility of exceedance for “Very Happy” of 0.80/1.0. That is, it is highly likely that “John” is not “Very Happy.”

The top graph in Figure 1-11 is the belief/plausibility distribution for “Happiness” for “John.” It is similar to the probability density distribution for a discrete random (numerical) variable,

---

3 The likelihood of exceedance for a fuzzy set is the likelihood that the variable of interest is worse than the indicated fuzzy set. That is, in the lower graph of Figure 1-11, the belief/plausibility of exceedance for “Very Happy” is the belief/plausibility for “Happiness” being worse than “Very Happy.” The belief/plausibility of exceedance for “Very Happy” is calculated as the belief/plausibility for {“Accepting, “Depressed”}. 

19
except the values are fuzzy sets, not numbers, and the measure of likelihood is a belief/plausibility interval, not a point probability. The bottom graph in Figure 1-11 is the complementary cumulative belief/plausibility distribution for “Happiness” for “John.” It is similar to the complementary cumulative probability distribution for a discrete random (numerical) variable, except the values are fuzzy sets, not numbers, and the measure of likelihood is a belief/plausibility interval, not a point probability.

An individual other than “John” can be evaluated by modifying the focal elements in the model to reflect the appropriate evidence. For example, suppose the information available for “Janet” yields the following focal elements. For “Health:”
- 0.3 to {“Moderate”}, and
- 0.7 to {“Moderate,” “Excellent”}.
For “Wealth”:
- 0.9 to {“Middle Class”}, and
- 0.1 to {“Middle Class,” “Rich”}.
For “Outlook on Life”:
- {“Optimist”} with evidence 0.85, and
- {“Pessimist,”} with evidence 0.15.

Figures 1-12 and 1-13 show the “Happiness” for “Janet.”

---

4 If all the focal elements are singletons, each belief/plausibility interval degenerates into a point that is the probability. In this situation the results are probability distributions over fuzzy sets. Section 4.1 provides an example.
Figure 1-13. Graphical Results for “Happiness” for “Janet”

It is likely that “Janet” is “Very Happy” or “Accepting” since the belief/plausibility for exceedance of “Accepting” is only 0.04/0.15.

Using plausibility, the results imply that “John” is less happy than “Janet.” The plausibility that “John” is “Depressed” is 1.0, and the plausibility that “Janet” is “Depressed” is 0.15.5

---

5 Section 3.2 provides an algorithm for ranking results using both plausibility and belief, and this example is evaluated using that algorithm in Section 3.2.
2 Using LinguisticBelief

Section 2.1 summarizes the top-level structure of the LinguisticBelief gui and Section 2.3 shows how to create an analysis in LinguisticBelief.

2.1 Structure of LinguisticBelief Graphical User Interface

Figure 2-1 is the frame (JFrame) for LinguisticBelief for a New analysis. A New analysis is an empty analysis, and is created by starting the application or by selecting either the New button or the File|New menu item.

![Figure 2-1. A New Analysis in LinguisticBelief](image)

2.2 Analysis Structure

Figure 2-2 shows the frame containing the analysis discussed in Section 1. The frame has:

- a menu bar (File, Utilities, Help)
- a button bar (New, Open, Save, Exit)
- two panels (JPanels):
  - The left panel displays the linguistics for the current analysis in a tree form (JTree),
  - the right panel displays information on the state of a linguistic variable selected in the tree in the left panel.
In Figure 2-2, the “Quality of Life” linguistic variable is selected in the tree in the left panel. The information on the current state of “Quality of Life” is provided in the right panel.

### 2.2.1 Commonly Used Options

The following commonly used options are provided on the button bar:
- **New**—Creates an empty analysis
- **Open**—Loads an old analysis previously saved in xml format
- **Save**—Saves the current analysis in extended markup language (xml) format
- **Exit**—Closes the application

All of these options, and other less frequently used options (**Save As** and **Delete**), are available under the **File** menu as shown in Figure 2-3. Other options include:
- **Save As**—Saves the current analysis in xml format under a different user-specified name, allowing further modifications to be made to the newly-named analysis without affecting the state of the prior analysis.
- **Delete**—Allows the user to remove an analysis from the collection of saved xml analyses

---

6 Analyses can also be saved and loaded in Java serialized (binary) form using the Utilities menu as subsequently discussed. Typically, analyses are saved and opened in the xml format.
2.2.2 Help Menu Options

The Help menu provides the options shown in Figure 2-4.

![Figure 2-3. Options under the File Menu](image)

![Figure 2-4. Options under the Help Menu](image)
The information provided by each of these Help options is shown in Figures 2-5 through 2-7.

![About LinguisticBelief](image1)

**Figure 2-5. The Menu Item Help|About**

![Limitations on an Analysis in LinguisticBelief](image2)

**Figure 2-6. The Menu Item Help|Analysis Limitations**

The “Help|Analysis Limitations” dialog shown in Figure 2-6 summarizes the limitations enforced by exception handlers in the code. For example, each linguistic variable must have a unique name, and each rule-based linguistic must have at least two and no more than four input linguistics.²

² The limitation of four inputs in the gui is a practical one, based on the difficulty of the user assigning rules with combinations of more than four inputs. Any linguistic variable requiring more than four inputs can be modified to have no more than four inputs, the inputs themselves having inputs as necessary to reflect the desired reasoning rule base. The algorithm that evaluates inputs is not restricted to four inputs; it is recursive and can accept any number of inputs as discussed in Appendix A. The four input limitation is on the gui. In the future, rule linguistics with more than four inputs may be input from other tools and can be handled by the algorithm in LinguisticBelief.
The following changes to an Analysis are propagated recursively throughout all Rule Linguistics. (1) Deletion of a selected Linguistic (Basic or Rule) does the following to every Rule Linguistic that uses the selected Linguistic as an input Linguistic: removes all focal elements, removes all rules, and deletes the selected Linguistic as an input. (2) Changing the focal elements for a Basic Linguistic removes all focal elements from any Rule Linguistic that uses that Basic Linguistic. (3) Renaming a Linguistic (Basic or Rule) only changes the name of that linguistic wherever it is used as an input linguistic to a Rule Linguistic. (4) Renaming a Fuzzy Set only changes the name of that fuzzy set wherever it is used as an input linguistic to a Rule Linguistic. (5) Adding or deleting a Fuzzy Set from a Linguistic (Basic or Rule) does the following to every Rule Linguistic that uses the selected Linguistic as an input Linguistic: removes all focal elements, and removes all rules. (6) Changing the input linguistics to a Rule Linguistic does the following to every Rule Linguistic that uses the selected Linguistic as an input Linguistic: removes all focal elements, and removes all rules.

Figure 2-7. The Menu Item Help|Changes to an Analysis

The “Help| Changes to an Analysis” dialog in Figure 2-7 describes how changes to an analysis change the state of that analysis. LinguisticBelief recursively propagates any change to an analysis throughout the entire analysis, saving portions of the analysis not affected by the change. For example, deletion of a linguistic variable removes that variable and removes all rules for any rule-based linguistic that requires the deleted variable as an input. Changing the name of a linguistic variable does not change the rules that use that variable.
2.2.3 Utilities Menu

The items under the Utilities menu are shown in Figure 2-8.

![Figure 2-8. Options Under the Utilities Menu](image)

The **Utilities|Help** choice displays information describing the other options under Utilities, as shown in Figure 2-9.

![Figure 2-9. The Menu Item Utilities|Help](image)
As indicated in Figure 2-9, Utilities options include:

- **Utilities|SaveSerialized** saves the current analysis in Java serialized (binary) form
- **Utilities|OpenSerialized** loads an existing analysis that was saved in serialized form.
- **Utilities|DeleteSerialized** deletes an existing analysis that was saved in serialized form.

As discussed in Appendix A, LinguisticBelief has the capability to save and load an analysis in serialized form as well as in xml format. The **Save**, **SaveAs**, and **Open** options previously discussed use the xml format.

- Check **Utilities|DebugOn** to provide information dialogs and dump detailed information to the console window (System.out). For example, with DebugOn selected, if the user tries to delete the linguistic variable “Quality of Life” for the example analysis in Section 1, the code provides the warning shown in Figure 2-10.

If the user selects “Yes” in the dialog in Figure 2-10, the code will delete the rule-based linguistic “Quality of Life” and dump the information shown in Listing 2-1 to the console. (Note the effect of recursion as discussed in Appendix A.)

**Listing 2-1. Example of Information Provided when Removing a Linguistic Variable with Utilities|DebugOn Selected**

```
RuleLinguisticDialog is deleting Quality of Life from RuleLinguistic Quality of Life
deleteThisAndAllRulesAndInputLinguisticIfInputLinguistic() looking at: Quality of Life

RuleLinguisticDialog is deleting Quality of Life from RuleLinguistic Happiness
deleteThisAndAllRulesAndInputLinguisticIfInputLinguistic() looking at: Happiness
  deleteThisAndAllRulesAndInputLinguisticIfInputLinguistic() operating recursively on: Quality of Life
deleteAllRulesAndInputLinguisticIfInputLinguistic() looking at: Quality of Life
  deleteThisAndAllRulesAndInputLinguisticIfInputLinguistic() removing Quality of Life
  deleteThisAndAllRulesAndInputLinguisticIfInputLinguistic() removed focal elements from Happiness
deleteThisAndAllRulesAndInputLinguisticIfInputLinguistic() removed rules from Happiness
```
2.3 Creating an Analysis

This section shows how to create an analysis by example, using the same one as discussed in Section 1.1. The procedure to create a New analysis follows:

- Click on the New button.
- Enter the name of the New analysis. If the user types the same name as the name of a previously saved analysis, the user has the option to replace the older analysis with the new one. Assuming that a new analysis named “Section 1 Example” is created, the gui appears as in Figure 2-11; an analysis named “Section 1 Example” has been created, with no linguistic variables as yet.

![Figure 2-11. “Section 1 Example” with No Linguistic Variables](image)

A basic linguistic is a linguistic to which evidence will be assigned. A rule-based linguistic is a linguistic formed from the combination of two or more other linguistics (basic or rule-based). LinguisticBelief requires that the name for every linguistic, basic or rule-based, be unique.

- To add a basic linguistic, left-click the Basic Linguistics node in the tree and enter the name of the basic linguistic to be added.
- To add a rule-based linguistic, left-click the Rule Linguistics node in the tree and entering the name of the rule-based linguistic to be added.

Figure 2-12 shows the results for the “Section 1 Example” after adding the Basic Linguistics “Health,” “Wealth,” and “Outlook on Life.” Figure 2-13 shows the results for the “Section 1 Example” after adding the Rule Linguistics “Quality of Life” and “Happiness.”
The next step is to operate on the Rule Linguistics and the Basic Linguistics you have created:

- Left click a linguistic variable (basic or rule) to highlight it.
- Right click on the selected variable to display the operations allowed for that variable.

For example, Figure 2-14 shows the Basic Linguistic options available for “Health.” Figure 2-17 shows the Rule Linguistic options available for “Quality of Life.”
2.3.1 Adding Fuzzy Sets

Next, the user adds fuzzy sets for all variables, both basic and rule. Click on the Add a Fuzzy Set radio button as shown in Figure 2-14 for a basic linguistic and Figure 2-15 for a rule linguistic.

Figure 2-14. Options for the Basic Linguistic “Health”

Figure 2-15. Options for the Rule Linguistic “Quality of Life”

Figure 2-16 shows the dialog for adding fuzzy sets to “Health” after all the fuzzy sets have been added.

- Type each fuzzy set in the Name of New Fuzzy Set to Add edit box and clicks Add Fuzzy Set button. Type each one separately; do not type commas.
• Add fuzzy sets for all the other linguistic variables in a similar manner. The system displays the fuzzy sets as you add them.

![Figure 2-16. Fuzzy Sets for “Health”](image)

### 2.3.2 Specify Input Linguistics

Once all the fuzzy sets for each linguistic variable (basic and rule-based) have been added, the user will specify the input linguistics for each rule-based linguistic. For example, selecting and right-clicking on “Quality of Life” shows the options in Figure 2-17.

![Figure 2-17. Specifying Input Linguistics for the Rule Linguistic “Quality of Life”](image)
Selecting the **Specify Input Linguistics** button (as indicated in Figure 2-17), a dialog displays. The user clicks the rule and basic variables needed as inputs. As the user clicks a variable, the system moves the variables to the top panel automatically. Figure 2-18 shows this dialog for “Quality of Life” before the input linguistics have been specified and Figure 2-19 shows this dialog after the appropriate input linguistics have been clicked.

![Figure 2-18. Dialog for Adding Input Linguistics to “Quality of Life”](image)

![Figure 2-19. “Quality of Life” with Input Linguistics Specified](image)
Similarly, the user selects input linguistics for “Happiness” as shown in Figure 2-20.

![Image of LinguisticBelief interface showing input linguistics for Happiness](image1.png)

**Figure 2-20. “Happiness” with Input Linguistics Specified**

If we try to add “Happiness” as another input linguistic for “Quality of Life” we get the error message shown in Figure 2-21, because “Happiness” itself has “Quality of Life” as an input and we cannot have circular logic.

![Image of error message showing circular logic cannot be detected](image2.png)

**Figure 2-21. Error Message if Circular Logic Is Detected**
2.4 Creating the Rule Base

Once all the basic and rule-based linguistics have been added, fuzzy sets added, and inputs for rule-based linguistics specified, the user will create rules for all the rule-based linguistics. Figure 2-22 shows the dialog for adding rules to “Quality of Life.” Note that all the “Output Fuzzy Set for Rule (blank if rule not set)” entries are blank, indicating that no rules have been specified. Figure 2-23 shows this dialog after all the rules have been added. To add an Output Fuzzy Set for Rule:

- Click on a blank line in the Output Fuzzy Set for Rule panel. (You may use standard Windows rules to select multiple lines to populate with the same selection.)
- Click on the combo box (Specify Output Fuzzy Set for Selected Rule) to display the list of fuzzy sets and select one. The blank line will be populated with your selection.
- When you are done, click on Accept Rules as Shown. You may accept a partial rule set, but all rules must be completed for a rule linguistic before you can calculate the result for that linguistic (Refer to Section 2.6.)
- If you click Cancel, you can revert to the previous rule set state without saving any changes since the last time you clicked Accept Rules as Shown.

![Figure 2-22. Empty Rule Base for ‘Quality of Life’](image)

![Figure 2-23. Complete Rule Base for “Quality of Life”](image)
Similarly, the user adds the rules for “Happiness” as shown in Figure 2-24.

![Figure 2-24. Complete Rule Base for “Happiness”](image)

### 2.5 Add Focal Elements

At this stage, the “Section 1 Example” model is complete and ready for use. To perform the evaluation for a specific situation, the evidence for that situation—the focal elements—must be specified. For our example analysis, Section 1.1 provided the focal elements for evaluating a specific individual, “John.” **Note:** The specific person’s name is not entered by the user in the application. To specifically identify any person (i.e., “John”), you can use the **Save As** option to copy the model under the name “John.” Now you change the evidence on the copied model “John” as discussed in this section. This allows you to separate your model from the specific situations to be evaluated.

To add focal elements:

- Right-click on a specific basic linguistic (Figure 2-13), e.g., “Health.”
- Select the **Set Focal Elements** button (Figure 2-14) to display a dialog that allows the user to add focal elements.
- Select one or fuzzy sets from the top panel and enter the evidence for the selected sets in the **Set the Evidence for the Focal Element** edit box. Repeat this step for each focal element (the combined evidence for all focal elements must sum to “1”). Figure 2-25 shows the dialog for adding the focal elements for “Health” where one focal element has been added: {“Bad,” “Moderate”} with evidence 0.8 and another focal element: {“Moderate,” “Excellent”} with evidence 0.2 is in the process of being added.
- After all focal elements are added, click **Cancel** to exit.
If the user clicks on **Cancel** before adding elements whose evidence sums to 1.0, the warning dialog in Figure 2-26 displays.

**Figure 2-26. Warning of Incomplete Evidence for a Basic Linguistic**

Figure 2-27 shows the result of adding all the focal elements to “Health.”
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Figure 2-27. All Focal Elements Added for “Health” for “John”

Should the user attempt to add another focal element to “Health,” the error dialog in Figure 2-28 is displayed as “Health” already has a set of focal elements whose evidence sums to 1.0.

Figure 2-28. Error Message for Attempt to Add a Focal Element that Would Result in Sum of Evidence over All Focal Elements Greater than One

To complete the body of evidence for “John,” the focal elements for “Wealth” and “Outlook on Life” given in Section 1.1 are also added using the same process just discussed for adding the focal elements to “Health.”

2.6 Calculate Results

Now, the results for any basic or rule linguistic for the specific situation can be calculated.

- Select and click any linguistic (basic or rule).
- Select Calculate Belief/Plausibility button (Figure 2-29).
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

For example, to calculate the results for “Quality of Life” for “John:”
- Select and right-click the “Quality of Life” node (Figure 2-13), resulting in the dialog shown in Figure 2-29.
- Select **Calculate Belief/Plausibility** button to display the results shown in Figure 2-30.

![Figure 2-29. Generating Results for “Quality of Life” for “John”](image)

![Figure 2-30. Results for “Quality of Life” for “John”](image)
In the results dialog of Figure 2-30, there is a button “Calculate [Belief, Plausibility] Interval for Selected Family of Fuzzy Sets.” Figure 2-31 shows an example use of this capability for the selected family {“Not so Good,” “Good”}; since this family is the entire universe of discourse for “Quality of Life,” both belief and plausibility are 1.0, as expected.

Figure 2-31. “Quality of Life” for “John” with a Family Selected

- Click on the Plot Results button shown in the dialog in Figure 2-31, to display the dialog in Figure 2-32, requiring the user to order the fuzzy sets from “Best” to “Worst.”

Figure 2-32. Fuzzy Sets, Not Ordered
• Click on the “Best” unordered fuzzy set in the lower panel. The fuzzy sets in the lower panel titled “Fuzzy Sets: Unordered “Best” to “Worst”” must be selected one at a time from “Best” to “Worst” and added to the upper panel titled “Fuzzy Sets: Ordered “Best” to “Worst”.”
• Click on the Order Selected Fuzzy Set button. The selected will move from the “Unordered” panel to the “Ordered” panel.

Figure 2-33 shows the results of adding “Good,” the “Best” fuzzy set, to the panel of ordered fuzzy sets.

![Figure 2-33. Fuzzy Sets, Partially Ordered](image1)

Figure 2-34 shows the results after all the fuzzy sets have been ordered from “Best” to “Worst.”

![Figure 2-34. Fuzzy Sets, All Ordered](image2)
Once all the fuzzy sets have been ordered, click the **Show Plots** button in Figure 2-34 to display the graphical results of Figure 2-35.

![Graphical Results for “Quality of Life” for “John”](image)

**Figure 2-35. Graphical Results for “Quality of Life” for “John”**

Similarly, the results for any of the other Rule Linguistics in the analysis can be generated. Figures 2-36 through 2-38 show results for “Happiness.”
Figure 2-36. “Happiness” for “John”

Figure 2-37. “Happiness” Fuzzy Sets Ordered “Best” to “Worst”
Figure 2-38. Graphical Results for “Happiness” for “John”
Linguistic Belief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Results for any linguistic (basic or rule) can be generated. Figures 2-39 through 2-41 show the results for “Wealth.”

**Figure 2-39. “Wealth” for “John”**

**Figure 2-40. “Wealth” Fuzzy Sets Ordered “Best” to “Worst”**
Figure 2-41. Graphical Results for “Wealth” for “John”


2.7 Saving an Analysis

The current analysis – “Section 1 Example” in this case – can be saved in xml format by clicking on the Save button on the frame button bar (shown, for example, on Figure 2-13). Using Utilities|SaveSerialized, the analysis can also be saved in Java serialized form as described in Appendix A.

2.8 Changing an Existing Analysis

Changes to an existing analysis may be made. For example, fuzzy sets and input linguistics for a rule-based linguistic can be changed as indicated in Figure 2-42. Fuzzy sets and focal elements for a basic linguistic can be changed as indicated in Figure 2-43.

![Figure 2-42. Dialog for Changing Fuzzy Sets and Input Linguistics for a RuleLinguistic](image)

![Figure 2-43. Dialog for Changing Fuzzy Sets and Focal Elements for a BasicLinguistic](image)


2.9 Creating Another Specific Situation

To evaluate another situation, (e.g., another person such as “Mary” for our example), you can select File|Save As to create a new file for Mary, using the existing model for “John.” Type a file name to represent the new person, “Mary,” and then change the evidence for John to apply to Mary, as explained in this section.

To delete all focal elements from the current model

- Left-click the top node in the tree; for example, the “Current Analysis: Section 1 Example” node in Figure 2-1.
- Right-click the highlighted selection to display the dialog in Figure 2-44.
- Click on Yes to remove all focal elements.
- Click on No if you do not want to remove all focal elements. You have cloned John and you may wish to change only a few focal elements, as discussed in Section 2.8.

![Figure 2-44. Dialog for Removing all Focal Elements](image)
3 A Complex Example

The prior sections used a simple example to illustrate the features of LinguisticBelief and to demonstrate how to use the tool.

This section briefly summarizes an application of LinguisticBelief for a real problem: evaluating and ranking the risk of terrorist scenarios.

3.1 An Example Model for Evaluating Terrorist Scenarios

Figure 3-1 shows the analysis model for this application. This model contains the linguistic variables and rule set for evaluating various terrorist scenarios as the Risk of the scenario.

Risk is a combination of Threat, Vulnerability, and Consequence. Threat is evaluated from the adversary perspective and includes the adversary perception of Vulnerability and Consequence. Specifically, Threat reasons on: Adversary Information Required, Adversary Attributes Required, and Adversary Estimate of Consequence. Vulnerability and Consequence are evaluated from the defender perspective. Vulnerability reasons on: Detect Adversary Gathering Resources and Defeat Attack. Consequence reasons on: Deaths, $ Loss, and Damage to National Morale.
Figure 3-2 shows the rule base created for the top Rule Linguistic “Risk.”
For one scenario of concern, Figure 3-3 shows the evaluation of Risk using dummy data for the focal elements for that scenario.

![Figure 3-3. Results for Risk for Scenario 1](image-url)
The graphical results for Risk for Scenario 1 are shown in Figure 3-4.

Figure 3-4. Graphical Results for Risk for Scenario 1
For another scenario of concern, Figure 3-5 shows the evaluation of Risk using dummy data for the focal elements for that scenario. Figure 3-6 shows the graphical results for Scenario 2.
3.2 Ranking Scenarios

For two scenarios, the graphs in the lower panels of Figures 3-4 and 3-6 summarize risk as a “likelihood” of exceedance of fuzzy sets, with the fuzzy sets ranked from least to most risk. These results can be used to rank the scenarios.

Let the fuzzy set $f_a$ be “worse” than the fuzzy set $f_b$. For any specific scenario “k:"

$$\text{Plaus}_{k \text{ exceed } f_a} \leq \text{Plaus}_{k \text{ exceed } f_b}$$

(Eqn. 1)

where $\text{Plaus}_{k \text{ exceed } f_a}$ is the plausibility of exceedance of fuzzy set $f_a$ for scenario “k.”

Scenarios are ranked by non-zero plausibility of exceeding the “worst” given fuzzy set (decreasing). For scenarios with equal ranking by plausibility, these scenarios are subranked by
belief of exceeding the fuzzy set (decreasing). Note that the ranking of the scenarios can be different for \( f_a \) than for \( f_b \).

Ranking of a few scenarios can be easily done manually; however, for a large number of scenarios the ranking must be automated. The ranking process has been implemented in a Java utility code, RankScenarios, written by the author. This code iterates over all fuzzy sets, worst to best, and for each fuzzy set ranks the scenarios by non-zero decreasing plausibility of exceeding that fuzzy set, and – for scenarios with the same plausibility – subranks by decreasing belief of exceeding that fuzzy set.

For the two scenarios summarized in Figures 3-4 and 3-6, the ranking results from the RankScenarios code are as given in Listing 3-1.

**Listing 3-1. Results from RankScenarios for Two Terrorist Scenarios**

**RANKING FOR SCENARIOS 1 and 2**

For Exceeding Fuzzy Set High the Scenarios rank ordered (decreasing) are:
Scenario 2 has plausibility of exceedance of 0.24 and belief of exceedance of 0.0

For Exceeding Fuzzy Set Moderate the Scenarios rank ordered (decreasing) (not already ranked for a worse fuzzy set) are:
(null)

For Exceeding Fuzzy Set Low the Scenarios rank ordered (decreasing) (not already ranked for a worse fuzzy set) are:
Scenario 1 has plausibility of exceedance of 0.03 and belief of exceedance of 0.0

For Exceeding Fuzzy Set Negligible the Scenarios rank ordered (decreasing) (not already ranked for a worse fuzzy set) are:
(null)

Figures 3-7 and 3-8 graphically display the ranking of these two scenarios. This type of graphical summary was proposed by John Cummings of SNL. Note that in these figures the fuzzy set name has been offset from the belief/plausibility interval for exceeding that fuzzy set to emphasize that these are exceedance plots. For example, in Figure 3-7, the belief/plausibility of exceeding “Negligible” Risk is 0/0.53, which is the belief/plausibility that Risk is either “Low,” “Moderate,” “High,” or “Very High.”
Risk for Scenario 1

Risk Fuzzy Sets

Belief/Plausibility Interval of Fuzzy Set

Belief  
Plausibility

"Likelihood" is bounded by Belief (lower bound) and Plausibility (upper bound).

Scenario is Ranked by Plausibility (upper bound) with subranking by Belief (lower bound).

Figure 3-7. Graphical Presentation for Ranking of Scenario 1

Risk for Scenario 2

Risk Fuzzy Sets

Belief/Plausibility Interval of Fuzzy Set

Belief  
Plausibility

"Likelihood" is bounded by Belief (lower bound) and Plausibility (upper bound).

Scenario is Ranked by Plausibility (upper bound) with subranking by Belief (lower bound).

For Scenario 2, "likelihood" of exceeding Low Risk is:
Plausibility 0.03
Belief 0

For Scenario 2, "likelihood" of exceeding High Risk is:
Plausibility 0.24
Belief 0

Figure 3-8. Graphical Presentation for Ranking of Scenario 2
The ranking technique presented in this section is not restricted to ranking terrorist scenarios. It can be used to rank results for any situation where the fuzzy sets for a variable of concern are ordered from “best” to “worst.” For example, Section 1 evaluated the “Happiness” of two individuals, “John” and “Janet,” and concluded that “John” is not as happy as “Janet.” This ranking can be formalized using the RiskScenarios code with the results given in Listing 3-2.

**Listing 3-2. Results from RankScenarios for Happiness**

---

**RANKING FOR JOHN AND JANET**

For Exceeding Fuzzy Set Accepting the Scenarios rank ordered (decreasing) (not already ranked for a worse fuzzy set) are:

- Scenario John has plausibility of exceedance of 1.0 and belief of exceedance of 0.016
- Scenario Janet has plausibility of exceedance of 0.15 and belief of exceedance of 0.041

For Exceeding Fuzzy Set Very Happy the Scenarios rank ordered (decreasing) (not already ranked for a worse fuzzy set) are:

- (null)

---

Figures 3-9 and 3-10 show these rankings graphically.
Figure 3-10. Graphical Presentation for Ranking of Happiness of Janet
4 Special Cases

This section illustrates two degenerate cases of belief/plausibility: probability and certainty. Also, the case with total ignorance is discussed.

All the examples in the earlier sections required use of the belief/plausibility measure of uncertainty because the evidence was non-specific. For example, for the “Health” of “John” we assigned the following focal elements in the example in Section 1:

- 0.8 to {“Bad,” “Moderate”}, and
- 0.2 to {“Moderate,” “Excellent”}.

Using this body of evidence, the results for “Health” for “John” are given in Figure 4-1.

---

Figure 4-1. Graphical Results for “Health” for “John”

---

8 The focal elements were not all singletons. [Darby]
Note that the belief or plausibility for exceedance of a fuzzy set cannot in general be directly calculated from the belief or plausibility values for the fuzzy sets that are worse than the fuzzy set of concern. For example, the plausibility of exceedance of “Excellent” (1.0 as shown in the lower graph of Figure 4-1) is not the sum of the plausibility of “Moderate” and “Bad” (1.0 + 0.8 as shown in the upper graph Figure 4-1). As discussed in a reference, belief and plausibility must be calculated using the focal elements. [Darby] Specifically, the belief (Bel) and plausibility (Pl) for any event A are:

\[
\begin{align*}
Bel(A) &= \sum_{B|B \subseteq A} m(B) \\
Pl(A) &= \sum_{B|A^c \cap B \neq \emptyset} m(B)
\end{align*}
\] (Eqn. 2)

where \(m(B)\) is the evidence assigned to event B (B is a focal element).

For the special case of probability, the probability of exceedance of a fuzzy set can be calculated directly as the sum of the probabilities for the fuzzy sets that are worse than the fuzzy set of concern. The next section provides an example where belief and plausibility are both probability.

4.1 Probability

If the evidence is specific enough, the focal elements are singletons and both belief/plausibility reduce to probability. That is, probability is a special case, or a degenerate case, of belief/plausibility. [Darby] For example, assume that for the “Health” of “Fred” we assign the following focal elements:

- 0.1 to {“Bad,”},
- 0.6 to {“Moderate,”}, and
- 0.3 to {“Excellent”}.

---

9 Necessity/possibility is another special case of belief/plausibility. [Darby] In this situation the focal elements are nested.
Using this body of evidence, the results for “Health” for “Fred” are given in Figure 4-2.

![Graphical Results for “Health” for “Fred”](image)

**Figure 4-2. Graphical Results for “Health” for “Fred”**

Note, in Figure 4-2 belief and plausibility both are a single value, which is the probability. Here, the belief/plausibility distribution is a probability distribution.

Note that the probability for exceedance of a fuzzy set can be directly calculated from the probability values for the fuzzy sets that are worse than the fuzzy set of concern. For example, the probability of exceedance of “Excellent” (0.7 as shown in the lower graph of Figure 4-2) is the sum of the probability of “Moderate” and “Bad” (0.6 + 0.1 as shown in the upper graph of Figure 4-2).

For probability, the focal elements are singletons; each focal element has a single fuzzy set. In Equation 2, if each B is a singleton, then Bel(A) and Pl(A) are the same: the Probability(A).
4.2 Certainty

Certainty is a special, or degenerate, case of probability. For example, assume we know the “Health” for “Richard” is “Moderate” based on information from his physician. For “Richard” we assign the following focal elements:

1.0 to {“Moderate”}.

Using this body of evidence, the results for “Health” for “Richard” are given in Figure 4-3.

Figure 4-3. Graphical Results for “Health” for “Richard”

For “Richard,” belief and plausibility both are a single value, probability, and we are certain that “Richard” has “Moderate” “Health,” so the probability of “Moderate” is 1.0.
4.3 Total Ignorance

If we have no information as to the “Health” of an individual, say “Mary,” we have total ignorance and the focal elements are as follows:

1.0 to {“Bad,” “Moderate,” “Excellent”}.

Using this body of evidence, the results for “Health” for “Mary” are given in Figure 4-4.

There is no (zero) belief for any of the fuzzy sets for the “Health” of “Mary,” but it is totally plausible (one) that her “Health” can be “Bad,” “Moderate,” or “Excellent.”
5 Summary

This report summarizes the use of the LinguisticBelief Java code for performing linguistic evaluations. Uncertainty is considered by segregating each linguistic variable into fuzzy sets. Linguistic variables are combined using an approximate reasoning rule base on the fuzzy sets. Uncertainty is captured and propagated using the belief/plausibility measure of uncertainty, thereby allowing considerable epistemic uncertainty to be considered. For situations where the uncertainty is purely aleatory, both belief and plausibility become the traditional probability measure of uncertainty.
Appendix A. Code Architecture and Deployment

Section A.1 summarizes the architecture of LinguisticBelief. Section A.2 summarizes the deployment package for LinguisticBelief.

A.1 Summary of Code Architecture

Figure A-1 shows the structure of the LinguisticBelief application in the NetBeans Integrated Development Environment (IDE). As indicated in the Projects panel in the left side of the figure, the application is segregated into numerous Java packages. The “datapackage” contains the classes used to perform calculations and the “guipackage” contains the classes used for gui. The “xmlJavatools” package contains the classes used for saving results in an xml (extended markup language) format.

The tool allows an analysis to be saved in both xml format and Java serialized (binary) format. The xml files are humanly readable; the binary files are not. The coding in LinguisticBelief required to implement handling data in xml is extensive; LinguisticBelief implements xml using a SAX parser. The coding required to implement serialization of data is simple, since serialization is built into Java. The “xmlpackage” contains all analyses saved in xml format. The “serializedAnalyses” package contains all analyses saved using serialization. Listing A-1 is a portion of the xml data file for the model discussed in Section 1.1.

The “Libraries” package indicates the libraries used by the application, including the JFreeChart library. Figure A-2 shows the JFreeChart library in the IDE.

The other packages shown in Figure A-1 are not pertinent to this discussion.

As indicated in Figure A-1, the Driver class contains the main method for the application.

A basic linguistic is implemented by the class datapackage.BasicLinguistic. A rule-based linguistic is implemented by the class datapackage.RuleLinguistic, which extends BasicLinguistic; thus, a RuleLinguistic “is-a” BasicLinguistic; that is, BasicLinguistic is a super class for the subclass RuleLinguistic, and RuleLinguistic inherits all the properties of BasicLinguistic.

Java is multithreaded. Since Java Swing is not thread safe, the Driver.main method runs the gui in the event dispatch thread, as indicated in Figure A-1.

The class datapackage.LinguisticBeliefException extends the Java class Exception and is used to trap errors.

Numerous inner classes are used as appropriate; for example, Rule is an inner class declared within RuleLinguistic as shown in Figure A-3. Also, LinguisticBelief makes extensive use of generics (a feature new to Java as of JSDK5) for data structures. For example, as indicated in Figure A-3, RuleLinguistic has the data member

```java
private ArrayList<BasicLinguistic> theInputLinguistics;
```
that declares the InputLinguistics as a reference to an ArrayList that contains instances of BasicLinguistic as elements. (ArrayList is a data type built into Java that allows dynamic creation and manipulation of a collection of elements.) LinguisticBelief will not run with versions of Java earlier than JSDK5.

The “guipackage” uses numerous standard Java Swing gui classes. The top-level gui element is a JFrame; for example, Figure 1-2 in the main report is a JFrame. The frame uses a JTree to display and allow the user to modify an analysis; for example, the left-side panel (a JPanel) in Figure 1-2 contains a JTree.
Figure A-1. LinguisticBelief in the NetBeans IDE
Listing A-1. Portion of an Analysis Saved in xml Format

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE linguisticBeliefData SYSTEM "LinguisticBeliefXMLData.dtd">
<!-- this xml file was generated by Analysis.dumpAnalysisToXMLFile() -->
<linguisticBeliefData>
  <!-- Name Of Analysis -->
  <analysisName>Report Example</analysisName>

  <!-- all the BasicLinguistics used in Analysis -->
  <basicLinguistic>
    <basicLinguisticName>Health</basicLinguisticName>
    <basicLinguisticFuzzySet>Bad</basicLinguisticFuzzySet>
    <basicLinguisticFuzzySet>Moderate</basicLinguisticFuzzySet>
    <basicLinguisticFuzzySet>Excellent</basicLinguisticFuzzySet>
    <basicLinguisticFocalElement>
      <focalElementFuzzySet>Bad</focalElementFuzzySet>
      <focalElementFuzzySet>Moderate</focalElementFuzzySet>
      <evidence>0.800</evidence>
    </basicLinguisticFocalElement>
    <basicLinguisticFocalElement>
      <focalElementFuzzySet>Moderate</focalElementFuzzySet>
      <focalElementFuzzySet>Excellent</focalElementFuzzySet>
      <evidence>0.200</evidence>
    </basicLinguisticFocalElement>
  </basicLinguistic>

  <basicLinguistic>
    <basicLinguisticName>Wealth</basicLinguisticName>
    <basicLinguisticFuzzySet>Poor</basicLinguisticFuzzySet>
    <basicLinguisticFuzzySet>Middle Class</basicLinguisticFuzzySet>
    <basicLinguisticFuzzySet>Rich</basicLinguisticFuzzySet>
    <basicLinguisticFocalElement>
      <focalElementFuzzySet>Middle Class</focalElementFuzzySet>
      <evidence>0.300</evidence>
    </basicLinguisticFocalElement>
    <basicLinguisticFocalElement>
      <focalElementFuzzySet>Poor</focalElementFuzzySet>
      <focalElementFuzzySet>Middle Class</focalElementFuzzySet>
      <evidence>0.700</evidence>
    </basicLinguisticFocalElement>
  </basicLinguistic>

  <basicLinguistic>
    <basicLinguisticName>Outlook on Life</basicLinguisticName>
    <basicLinguisticFuzzySet>Pessimist</basicLinguisticFuzzySet>
    <basicLinguisticFuzzySet>Optimist</basicLinguisticFuzzySet>
    <basicLinguisticFocalElement>
      <focalElementFuzzySet>Pessimist</focalElementFuzzySet>
      <evidence>0.980</evidence>
    </basicLinguisticFocalElement>
    <basicLinguisticFocalElement>
      <focalElementFuzzySet>Pessimist</focalElementFuzzySet>
      <focalElementFuzzySet>Optimist</focalElementFuzzySet>
      <evidence>0.020</evidence>
    </basicLinguisticFocalElement>
  </basicLinguistic>
</linguisticBeliefData>
<ruleLinguistic>
<ruleLinguisticName>Quality of Life</ruleLinguisticName>
<ruleLinguisticFuzzySet>Good</ruleLinguisticFuzzySet>
<ruleLinguisticFuzzySet>Not so Good</ruleLinguisticFuzzySet>
<ruleLinguisticBasicLinguisticInput>Health</ruleLinguisticBasicLinguisticInput>
<ruleLinguisticBasicLinguisticInput>Wealth</ruleLinguisticBasicLinguisticInput>
<rule>
<ruleInput>
<ruleInputBasicLinguisticName>Health</ruleInputBasicLinguisticName>
<ruleInputBasicLinguisticFuzzySet>Bad</ruleInputBasicLinguisticFuzzySet>
</ruleInput>
<ruleInput>
<ruleInputBasicLinguisticName>Wealth</ruleInputBasicLinguisticName>
<ruleInputBasicLinguisticFuzzySet>Poor</ruleInputBasicLinguisticFuzzySet>
</ruleInput>
<ruleOutputFuzzySet>Not so Good</ruleOutputFuzzySet>
</rule>

(Rest of file not shown here)
Figure A-2. JFreeChart as a Library in the IDE
Figure A.3. Rule as an Inner Class in RuleLinguistic
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

LinguisticBelief uses recursion to effect changes to an analysis. For example, the gui class datapackage.RuleLinguistic uses the RuleLinguistic.deleteAllRulesAndInputLinguisticIfInputLinguistic() method in Listing A-2 to respond to a user request to delete a previously created RuleLinguistic.
/** recursively delete rules/focal elements if this has theInputLinguistic as an input linguistic 
* (but do not delete theInputLinguistic as an input linguistic for this since that is done in the gui calling method)
*/
public void deleteAllRulesAndInputLinguisticIfInputLinguistic(RuleLinguistic topRuleLinguistic, RuleLinguistic currentRuleLinguistic, BasicLinguistic theInputLinguisticToDelete) {
    // topRuleLinguistic is the RuleLinguistic of interest and is not changed in this method
    // currentRuleLinguistic is any RuleLinguistic that is an input into topRuleLinguistic
    // theInputLinguisticToDelete is the input linguistic to be deleted and is not changed in this method
    if ( Driver.isDebugOn() ) {            System.out.println("    deleteAllRulesAndInputLinguisticIfInputLinguistic() looking at: " + currentRuleLinguistic);
        Iterator<BasicLinguistic> iterInputLing = currentRuleLinguistic.getTheInputLinguistics().iterator();
        while (iterInputLing.hasNext() ) {             BasicLinguistic anInputLinguistic = iterInputLing.next();
            if ( anInputLinguistic.getClass() == RuleLinguistic.class ) { // this may have an input with an input that is theInputLinguistic
                if ( ( (RuleLinguistic) anInputLinguistic).getTheFocalElements().isEmpty() ) { // anInputLinguistic may have had focal elements removed
                    if ( Driver.isDebugOn() ) {                         System.out.println("        deleteAllRulesAndInputLinguisticIfInputLinguistic() has a RuleLinguistic: " + anInputLinguistic +
                                " with no focal elements.  So, remove rules and focal elements from: " + topRuleLinguistic);
                    }
                    topRuleLinguistic.theRules = new ArrayList<Rule> ();
                    topRuleLinguistic.emptyFocalElements();
                    topRuleLinguistic.emptyBeliefPlausibility();
                }
            } else if (anInputLinguistic.equals(theInputLinguisticToDelete)) { // theInputLinguistic itself is an input for this
                if ( Driver.isDebugOn() ) {
                    System.out.println("        deleteAllRulesAndInputLinguisticIfInputLinguistic() operating recursively on: " + anInputLinguistic);
                }
                (RuleLinguistic) anInputLinguistic ).deleteAllRulesAndInputLinguisticIfInputLinguistic(topRuleLinguistic, (RuleLinguistic) anInputLinguistic, theInputLinguisticToDelete); // recursive call
            }
        }
    }
}
deleteAllRulesAndInputLinguisticIfInputLinguistic() removed focal elements from " + topRuleLinguistic);
} // remove rules from topRuleLinguistic
    topRuleLinguistic.emptyFocalElements();
    topRuleLinguistic.emptyBeliefPlausibility();
    if ( Driver.isDebugOn() ) {
        System.out.println(" deleteAllRulesAndInputLinguisticIfInputLinguistic() removed rules from " + topRuleLinguistic);
    }

    return;
} // if
} // end while iterInputLing
As discussed in a reference, the code performs an automatic “aggregation” of focal elements as it progresses through the evaluation of rule-based linguistic variables. [Darby]

This aggregation is best discussed by an example. Consider the rule base for the linguistics “Quality Of Life” and “Happiness” presented in Section 1.1. Using the LinguisticBelief Code the following results are obtained (selected output presented) with Utilities|DebugOn selected.

**Quality of Life Focal Elements**

For RuleLinguistic Quality of Life a FocalElement with degree of evidence 2.4000e-01 is:
- Bad & Middle Class
- Moderate & Middle Class

For RuleLinguistic Quality of Life a FocalElement with degree of evidence 5.6000e-01 is:
- Bad & Poor
- Moderate & Poor
- Moderate & Middle Class

For RuleLinguistic Quality of Life a FocalElement with degree of evidence 6.0000e-02 is:
- Moderate & Middle Class
  - Excellent & Middle Class

For RuleLinguistic Quality of Life a FocalElement with degree of evidence 1.4000e-01 is:
- Moderate & Poor
  - Moderate & Middle Class
  - Excellent & Poor
  - Excellent & Middle Class

**Quality of Life Rule Base**

For RuleLinguistic Quality of Life the fuzzy sets from union of rules with output FuzzySet Not So Good are:
- Bad & Poor
  - Moderate & Poor
  - Bad & Middle Class
  - Moderate & Middle Class
  - Bad & Rich

For RuleLinguistic Quality of Life the fuzzy sets from union of rules with output FuzzySet Good are:
- Excellent & Poor
  - Excellent & Middle Class
  - Moderate & Rich
  - Excellent & Rich

**Quality of Life Belief / Plausibility Intervals**

For BasicLinguistic Quality of Life For fuzzy set Not So Good Belief / Plausibility interval is: 8.00000e-01 / 1.00000e+00
For BasicLinguistic Quality of Life For fuzzy set Good Belief / Plausibility interval is: 0.00000e+00 / 2.00000e-01
Outlook on Life Belief / Plausibility Intervals

For BasicLinguistic Outlook On Life For fuzzy set Pessimist Belief / Plausibility interval is 2.00000e-02, 1.00000e+00
For BasicLinguistic Outlook On Life For fuzzy set Optimist Belief / Plausibility interval is 0.00000e+00, 9.80000e-01

To evaluate “Happiness” using the rule base, its focal elements need to be expressed in terms of the fuzzy sets of its input variables, which is accomplished by aggregation. For example, before aggregation the focal elements for “Happiness” are:

(1) For RuleLinguistic Happiness a FocalElement with degree of evidence: 4.8000e-03 is
Bad & Middle Class & Pessimist
Moderate & Middle Class & Pessimist

(2) For RuleLinguistic Happiness a FocalElement with degree of evidence: 2.3520e-01 is
Bad & Middle Class & Pessimist
Bad & Middle Class & Optimist
Moderate & Middle Class & Pessimist
Moderate & Middle Class & Optimist

(3) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.1200e-02 is:
Bad & Poor & Pessimist
Bad & Middle Class & Pessimist
Moderate & Poor & Pessimist
Moderate & Middle Class & Pessimist

(4) For RuleLinguistic Happiness a FocalElement with degree of evidence 5.4880e-01 is:
Bad & Poor & Pessimist
Bad & Poor & Optimist
Bad & Middle Class & Pessimist
Bad & Middle Class & Optimist
Moderate & Poor & Pessimist
Moderate & Poor & Optimist
Moderate & Middle Class & Pessimist
Moderate & Middle Class & Optimist

(5) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.2000e-03 is:
Moderate & Middle Class & Pessimist
Excellent & Middle Class & Pessimist

(6) For RuleLinguistic Happiness a FocalElement with degree of evidence 5.8800e-02 is:
Moderate & Middle Class & Pessimist
Excellent & Middle Class & Optimist
Excellent & Middle Class & Pessimist

(7) For RuleLinguistic Happiness a FocalElement with degree of evidence 2.8000e-03 is:
Moderate & Poor & Pessimist
Moderate & Middle Class & Pessimist
Excellent & Poor & Pessimist
Excellent & Middle Class & Pessimist

(8) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.3720e-01 is:
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Moderate & Poor & Pessimist
Moderate & Poor & Optimist
Moderate & Middle Class & Pessimist
Moderate & Middle Class & Optimist
Excellent & Poor & Pessimist
Excellent & Poor & Optimist
Excellent & Middle Class & Pessimist
Excellent & Middle Class & Optimist

The rule base for “Happiness” is:

For RuleLinguistic Happiness the fuzzy sets from union of rules with output FuzzySet Very Happy are:
Good & Optimist

For RuleLinguistic Happiness the fuzzy sets from union of rules with output FuzzySet Accepting are:
Not So Good & Optimist
Good & Pessimist

For RuleLinguistic Happiness the fuzzy sets from union of rules with output FuzzySet Depressed are:
Not So Good & Pessimist

The focal elements for “Happiness” contain “Quality Of Life” in terms of the fuzzy sets of its input variables “Health” and “Wealth,” but the rule base for “Happiness” is expressed in terms of the fuzzy sets for “Quality Of Life.” The focal elements for “Happiness” must be aggregated to be expressed in terms of the fuzzy sets of “Quality Of Life” instead of being expressed in terms of the fuzzy sets for the input variables for “Quality Of Life.”

Aggregation is performed in two steps. First, for any given focal element, its constituent input fuzzy sets are mapped to the appropriate output fuzzy set per the rule base. For “Happiness,” this results in the following focal elements:

(1) For RuleLinguistic Happiness a FocalElement with degree of evidence 4.8000e-03 is:
Not So Good & Pessimist

(2) For RuleLinguistic Happiness a FocalElement with degree of evidence 2.3520e-01 is:
Not So Good & Pessimist
Not So Good & Optimist

(3) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.1200e-02 is:
Not So Good & Pessimist

(4) For RuleLinguistic Happiness a FocalElement with degree of evidence: 5.4880e-01 is
Not So Good & Pessimist
Not So Good & Optimist

(5) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.2000e-03 is:
Not So Good & Pessimist
Good & Pessimist

(6) For RuleLinguistic Happiness a FocalElement with degree of evidence 5.8800e-02 is:
Not So Good & Pessimist
Not So Good & Optimist
Good & Pessimist
(7) For RuleLinguistic Happiness a FocalElement with degree of evidence 2.8000e-03 is:
   Not So Good & Pessimist
   Good & Pessimist

(8) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.3720e-01 is:
   Not So Good & Pessimist
   Not So Good & Optimist
   Good & Pessimist
   Good & Optimist

Second, focal elements resulting from step one containing identical fuzzy sets are combined by adding the degrees of evidence, resulting in the following focal elements for “Happiness.”

(A) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.6000e-02 is:
   Not So Good & Pessimist

(B) For RuleLinguistic Happiness a FocalElement with degree of evidence 7.8400e-01 is:
   Not So Good & Pessimist
   Not So Good & Optimist

(C) For RuleLinguistic Happiness a FocalElement with degree of evidence 4.0000e-03 is:
   Not So Good & Pessimist
   Good & Pessimist

(D) For RuleLinguistic Happiness a FocalElement with degree of evidence 1.9600e-01 is:
   Not So Good & Pessimist
   Not So Good & Optimist
   Good & Pessimist
   Good & Optimist

This two-step aggregation operation reduced the number of focal elements for “Happiness” from eight to four.

The most complex algorithm in the code is associated with calculating the focal elements for a Rule Linguistic from the focal elements of its constituent input linguistics. This algorithm uses two nested recursive methods to allow any number of input linguistics to be evaluated. Listing A-3 is the RuleLinguistic.calculateFocalElements() method that implements that algorithm.
/** RuleLinguistic.calculateFocalElements(): calculate focal elements assuming noninteraction */

public void calculateFocalElements() {

    // Consolidate focal elements for each input that is a RuleLinguistic.
    // For each input that is a BasicLinguistic consolidate does nothing
    Iterator<BasicLinguistic> iterInputLinguistics = this.theInputLinguistics.iterator();
    while (iterInputLinguistics.hasNext()) {
        BasicLinguistic inputLinguistic = iterInputLinguistics.next();
        inputLinguistic.consolidateFocalElements();
    }

    // form this.theFocalElements from each this.theInputLinguistics focal elements
    // allow any number of input linguistics using recursion
    // create a collection to hold a collection of FocalElements for each input linguistic
    ArrayList<ArrayList<FocalElement>> collectionInputFocalElements = new ArrayList<ArrayList<FocalElement>>();
    // fill collectionInputFocalElements
    Iterator<BasicLinguistic> iterInputs = this.theInputLinguistics.iterator();
    while (iterInputs.hasNext()) {
        collectionInputFocalElements.add(iterInputs.next().getTheFocalElements());
    }

    if ( Driver.isDebugOn() ) {
        System.out.println("the RuleLinguistic " + this + " called RiskLinguistic.calculateFocalElements() and set collectionInputFocalElements to be: " + collectionInputFocalElements);
    }

    // EXAMPLE this.RuleLinguistic, call it D, has 3 basicLinguistic inputs A, B, anc C
    // A has focal elements A1 & A2, A1 with evidence 0.2, 0.8
    // B has focal elements B1 & B2, B2 with evidence 0.8, 0.2
    // C has focal elements C1, C2 with evidence 0.5, 0.5
    // collectionInputFocalElements for D is: [[A1, A1 & A2], [B2, B1 & B2], [C1, C2]]

    // operate on collectionInputFocalElements to calculate the focal elements for this
    // the result for EXAMPLE D will have 8 focal elements with the following fuzzy sets
    /*
    A1_&_B2_&_C1 with evidence 8.0000e-02
    A1_&_B2_&_C2 with evidence 8.0000e-02
    A1_&_B2_&_C1 & A2_&_B2_&_C1 with evidence 8.0000e-02
    A1_&_B1_&_C1 & A1_&_B2_&_C1 & A2_&_B1_&_C1 & A2_&_B2_&_C1 with evidence 8.0000e-02
    where _&_ denotes combinations of fuzzy sets from each ordered input linguistic
    and _&_ denotes combinations for the resulting this.RuleLinguistic
    */

    if ( Driver.isDebugOn() ) {
        
}
System.out.println(" RiskLinguistic.calculateFocalElements() is CALLING
formFocalElements() to operate on
collectionInputFocalElements");
}
formFocalElements(collectionInputFocalElements, 0); // formFocalElements() is recursive, and
internally formFocalElements()
// calls setFuzzySetsForOneFocalElement() which is also recursive
if ( Driver.isDebugOn() ) {
    System.out.println(" formFocalElements() RETURNED");
    System.out.println("RiskLinguistic.calculateFocalElements() RETURNED");
}

// set this RuleLinguistic focal elements
this.setTheFocalElements(this.theFocalElementsForThisRuleLinguistic);
if ( Driver.isDebugOn() ) {
    System.out.println();
    System.out.println("RuleLinguistic " + RuleLinguistic.this.getName() + " has " +
this.getTheFocalElements().size() +
    " focal elements as follows");
    Iterator<FocalElement> iterFocalElements = theFocalElementsForThisRuleLinguistic.iterator();
    while (iterFocalElements.hasNext()) {
        FocalElement aFocalElement = iterFocalElements.next();
        System.out.println(" For RuleLinguistic " + RuleLinguistic.this.getName() + " a FocalElement with
degree of evidence %.4e is: 
," 
            aFocalElement.getTheEvidence());
        Iterator<FuzzySet> iterFuzzySets = aFocalElement.getTheFuzzySets().iterator();
        while (iterFuzzySets.hasNext()) {
            FuzzySet aFuzzySet = iterFuzzySets.next();
            System.out.println("    " + aFuzzySet.getName());
        } // end inner while
    } // end outer while
} // end calculateFocalElements()

/* RuleLinguistic.formFocalElements() */
// used by this.calculateFocalElements() to allow any number of input linguistics for this.
// this.calculateFocalElements() passes in collectionInputFocalElements, a collection of
// focal elements for each input in order; e.g., index 0 element is focal elements for input 1, etc.
// Recursively operate on all input focal elements
// first call is with nIndexInput = 0
private void formFocalElements(ArrayList<ArrayList<FocalElement>> collectionInputFocalElements, int
nIndexInput) {
    // EXAMPLE collectioncollectionInputFocalElements for D is: [[A1, A1 & A2], [B2, B1 & B2], [C1, C2]]

    // if index is one more than number of inputs
    // so we are done since we have completed evaluation all collectionInputFocalElements
    if (nIndexInput == collectionInputFocalElements.size() ) {
        return;
    }
    if (nIndexInput == 0) {
// initialize variables to be calculated
this.theFocalElementsForThisRuleLinguistic = new ArrayList<FocalElement> ();
// FocalElements for this formed from FocalElements for inputs to this

// get the focal elements for input at nIndexInput
Iterator<FocalElement> iterAnInputFocalElements =
collectionInputFocalElements.get(nIndexInput).iterator();
while ( iterAnInputFocalElements.hasNext() ) {
  // this.formulatingFocalElement.size() is a private data member of RuleLinguistic initially set to
  // remove all elements from end back through nIndexInput from this.formulatingFocalElement to
  // reset for next iteration
  for (int nRemove = this.formulatingFocalElement.size() - 1; nRemove >= nIndexInput; nRemove--)
  {
    if ( Driver.isDebugOn() ) {
      System.out.println("        RESETTING this.formulatingFocalElement.size(): " +
this.formulatingFocalElement.size() + " + nIndexInput + " .... Removing input index: " + nRemove);
    }
    this.formulatingFocalElement.remove(nRemove);
  }
  FocalElement anInputFocalElement = iterAnInputFocalElements.next();
  // add to this.formulatingFocalElement
  this.formulatingFocalElement.add(anInputFocalElement);
  // create new focal element for this if nIndexInput is last element of collection
  if ( nIndexInput == collectionInputFocalElements.size() - 1 ) {
    // ASSERT this.formulatingFocalElement has size = collectionInputFocalElements.size()
    // and this.formulatingFocalElement contains focal elements from inputs with
    // index 'n-1' a focal element from input 'n'
    // form a focal element for this RuleLinguistic from the focal elements of the 'n' inputs
    if ( Driver.isDebugOn() ) {
      System.out.println("        formFocalElements() is CALLING
setFuzzySetsForOneFocalElement() with
      this.formulatingFocalElement: " + this.formulatingFocalElement);
    }
    this.setFuzzySetsForOneFocalElement(this.formulatingFocalElement, 0); // recursive
    if ( Driver.isDebugOn() ) {
      System.out.println("      setFuzzySetsForOneFocalElement() RETURNED");
    }
  }
  try {
    this.theFocalElementsForThisRuleLinguistic.add(new
FocalElement(this.aFocalElementForThisRuleLinguistic,
  this.calculatedDegreeOfEvidence));
  if ( Driver.isDebugOn() ) {
    System.out.println("      this.RuleLinguistic currently has the following focal element:");
    for (FocalElement focElem: this.theFocalElementsForThisRuleLinguistic)
System.out.println(" + focElem);
} catch (LinguisticBeliefException lbex) {
}

} // end if (nIndexInput == collectionInputFocalElements.size() - 1)
// END create new focal element for this if nIndexInput is last element of collection
// recursively continue with the focal elements at nIndexInput
formFocalElements(collectionInputFocalElements, nIndexInput + 1 );

} /// end while iterAnInputFocalElements

} // end formFocalElements

/* RuleLinguistic.setFuzzySetsForOneFocalElement() */
// used by this.formFocalElements() to operate in this.formulatingFocalElement
// Recursively operate on all elements of this.formulatingFocalElement
// first call is with nIndexInput = 0.
// collectionOneFocalElementFromInputs is a collection of focal elements that will form one
// focal element for this. collectionOneFocalElementFromInputs has one focal element from each input
to this,
// ordered; e.g. index 0 is a focal element from input 1.
// setFuzzySetsForOneFocalElement is called with formulatingFocalElement as arg for
// collectionOneFocalElementFromInputs.
// degreeOfEvidence is degree of evidence for the new fuzzy set for this previously calculated.
private void setFuzzySetsForOneFocalElement(ArrayList<FocalElement> formulatingFocalElement, int nIndexInput) {

// if index is one more than number of inputs
// so we are done since we have completed evaluation all formulatingFocalElement
if (nIndexInput == formulatingFocalElement.size() ) {
    return;
}

// if index is 0 this is first call so start a new collection of fuzzy sets that will be filled and added
// as a focal element to this
// also calculate the degreeOfEvidence for the focal element for this
if (nIndexInput == 0) {
    this.aFocalElementForThisRuleLinguistic = new ArrayList<FuzzySet> ();
    // calculate the degreeOfEvidence for the new focal element
    this.calculatedDegreeOfEvidence = 1.0; // degree of evidence for new focal element for this
    Iterator<FocalElement> iterFocalElem = formulatingFocalElement.iterator();
    while (iterFocalElem.hasNext() ) {
        this.calculatedDegreeOfEvidence *= iterFocalElem.next().getTheEvidence();
    } // while iterFocalElem

    // get a fuzzy set from the focal element at nIndexInput
    Iterator<FuzzySet> indexFocalElementFuzzySet =
        formulatingFocalElement.get(nIndexInput).getTheFuzzySets().iterator();
    while ( indexFocalElementFuzzySet.hasNext() ) {

LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

// remove all elements up to nIndexInput from this.fuzzySetsForOneFocalElement to reset for next iteration
for (int nRemove = this.fuzzySetsForOneFocalElement.size() - 1; nRemove >= nIndexInput; nRemove--) {
    if (Driver.isDebugOn()) {
        System.out.println("RESETTING this.fuzzySetsForOneFocalElement.size(): " +
        this.fuzzySetsForOneFocalElement.size() +
        " for nIndexInput: " + nIndexInput + ".:.Removing input index: " + nRemove);
    }
    this.fuzzySetsForOneFocalElement.remove(nRemove);
}

FuzzySet anInputFocalElementFuzzySet = indexFocalElementFuzzySet.next();
// add to this.fuzzySetsForOneFocalElement
this.fuzzySetsForOneFocalElement.add(anInputFocalElementFuzzySet);

if (nIndexInput == formulatingFocalElement.size() - 1) {
    // ASSERT this.fuzzySetsForOneFocalElement has size =
    // collectionOneFocalElementFromInputs.size()
    // and this.fuzzySetsForOneFocalElement contains fuzzy sets from inputs with
    // index 'n-1' a fuzzy set from input 'n'
    String nameForCombinedFuzzySets = new String();
    boolean bIsFirstInputFuzzySet = true;
    Iterator<FuzzySet> iterFuzzySet = this.fuzzySetsForOneFocalElement.iterator();
    while (iterFuzzySet.hasNext()) {
        if (bIsFirstInputFuzzySet) {
            nameForCombinedFuzzySets += iterFuzzySet.next().getName();
            bIsFirstInputFuzzySet = false;
        } else
            nameForCombinedFuzzySets += "&_" + iterFuzzySet.next().getName();
    }
    // end iterFuzzySet
    this.aFocalElementForThisRuleLinguistic.add(new FuzzySet(nameForCombinedFuzzySets));
    if (Driver.isDebugOn()) {
        System.out.println("Fuzzy Sets for this.aFocalElementForThisRuleLinguistic: ");
        for (FuzzySet fuzzSet: this.aFocalElementForThisRuleLinguistic)
            System.out.println(" " + fuzzSet);
    }
} // end if (nIndexInput == collectionOneFocalElementFromInputs.size() - 1)

// recursively continue with the focal elements at the next nIndexInput
this.setFuzzySetsForOneFocalElement(formulatingFocalElement, nIndexInput + 1);
}
} // end setFuzzySetsForOneFocalElement

// formulatingFocalElement is used by this.formFocalElements() to find combinations of focal elements
private ArrayList<FocalElement> formulatingFocalElement = new ArrayList<FocalElement> ();
A.2 Deployment of LinguisticBelief

Using the “Clean and Build Main Project” option in the IDE, a jar file for the LinguisticBelief application was created in the “dist” subdirectory of the project, as indicated in the project file description of Figure A-4. The IDE automatically includes all libraries used by the project in the “dist/lib” subdirectory of the project.
Figure A-4. Deployment Package for LinguisticBelief in the Netbeans IDE
In creating the LinguisticBelief.jar in the IDE, the xmlpackage and the serializedAnalyses packages (shown in Figure A-1) were excluded from the LinguisticBelief.jar file using the properties settings indicated in Figure A-5. These packages were manually added to the distributed package in the “src” subdirectory as shown in Figure A-6. This allows the client direct access to these packages where analyses are saved in xml and serialized form, respectively.

The deployed project is distributed under the name LinguisticBelief as indicated in Figure A-6. A recent Java runtime environment (jre1.5.0_08) is included in the deployment, so that the user will have JDK5 available as required to run the application. A batch file, LinguisticBeliefRunJAR.bat, is included for running the application in Windows XP using the deployed Java runtime environment.

Figure A-7 shows the files in the lib subdirectory of the deployment package.
LinguisticBelief: A Java Application for Linguistic Evaluation Using Belief, Fuzzy Sets, and Approximate Reasoning

Figure A-6. The LinguisticBelief Deployment Package

Figure A-7. Contents of LinguisticBelief.lib
If this package is copied to a client machine, the application can be executed by running the “LinguisticBeliefRunJAR.bat” file. The commands in “LinguisticBeliefRunJAR.bat” are shown in Figure A-8.

![LinguisticBeliefRunJAR.bat - Notepad](image)

**Figure A-8. Commands in LinguisticBeliefRunJAR.bat**

The distributed package contains the xmlpackage and serializedAnalyses subdirectories as indicated in Figure A-6. The xmlpackage contains the document type description (DTD) file for the client-generated xml files, LinguisticBeliefXMLData.dtd, which must **not** be deleted from the distributed package.
References


## Distribution

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Code</th>
<th>Name</th>
<th>Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>MS 0757</td>
<td>John Darby, 06462</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MS 0762</td>
<td>S. Jordan, 06407</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MS 9018</td>
<td>Central Technical Files, 08944</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MS 0899</td>
<td>Technical Library, 04536</td>
<td></td>
</tr>
</tbody>
</table>