Calculating safety and reliability probabilities with functions of uncertain variables can yield incorrect or misleading results if some precautions are not taken. One important consideration is the application of constrained mathematics for calculating probabilities for functions that contain repeated variables. This paper includes a description of the problem and develops a methodology for obtaining an accurate solution.

Introduction

Safety and reliability analyses often depend on Boolean logic combinations of input variables that have uncertainty (imperfect knowledge) or variability (probabilistically described outcomes). It is now widely accepted that such analyses must address several subtleties, including:

1. Converting the standard “sum of cutsets” Boolean expression to a probabilistic solution by the “rare event approximation” (arithmetic sum) can give misleading results, even for low-probability inputs, because of aggregation of significant results from large numbers of individually insignificant contributors. One remedy for this problem is to use a disjoint set procedure [1].

2. Using “Monte-Carlo-like” simulations to approximate probabilities for logical combinations of variables for which uncertainty or variability is characterized by probability distributions can misrepresent extreme values. One remedy is to use extreme-value analysis [2].

3. Using probability distributions to model knowledge uncertainty can significantly misrepresent extreme values. Fuzzy or possibilistic algebras help address this problem [3].

It has recently been recognized [4] that using interval-based computations such as interval arithmetic, and fuzzy or possibilistic mathematics in an “unconstrained” mode (applied by sequentially parsing equation solutions) can significantly misrepresent extreme values. This phenomenon, its ramifications, and a solution for the problem will be discussed in this paper. Since risk management decisions are commonly based on the results of such analyses, the goal is to derive the most accurate bounds possible, so the resultant decisions can be as good as possible.
The “Repeated Variable” Problem in Risk Management

Mathematically processing uncertain operands may involve constraints inherent in the problem definition, such that computations are difficult to implement. An example of a possible constraint is that repeated appearances of the same uncertain variable must all have the identical value. This “repeated variable problem” will be addressed in order to show how range-based probabilistic evaluation of logic expressions, such as those describing the outcomes of fault trees and event trees, can be facilitated. The results are applicable to fuzzy or possibilistic mathematics, interval analysis, Monte Carlo or LHS analysis and other range-based techniques. The problem is important, because unconstrained computations result in wider ranges of outputs than those properly obtained with constrained mathematics. Since risk management decisions may be based on this information, the results should contain no errors.

We illustrate techniques that can be used to transform complex constrained problems into trivial problems in most tree logic expressions, and into tractable problems in most other cases. The approach is based on the Boolean logic characteristics of “unateness” [5], “minimal compactness” [6], and differential calculus characteristics related to regions of monotonicity. Example problems are used to demonstrate the techniques and the advantages of constrained mathematics. The results obtained are the precise bounds sought for risk management.

Constrained Mathematics

An example illustrating the necessity for constrained mathematics is the union of two independent events for which the probabilities are specified by intervals, \( P(A) = [x_l, x_u] \) and \( P(B) = [y_l, y_u] \). Using unconstrained operations of interval analysis, one obtains:

\[
P(A \cup B) = [x_l + y_l - x_u y_u, x_u + y_u - x_l y_l],
\]

for which both operands can give values outside the range \([0, 1]\). The error in unconstrained mathematics analysis of this problem is that both interval bounds derived from unconstrained operations combine an operand’s lower bound together with its upper bound, which is not possible for uncertainty about an event. Restricting events to have only one value at a time, one obtains the correct answer, which is:

\[
P(A \cup B) = [x_l + y_l - x_l y_l, x_u + y_u - x_u y_u]
\]

Unfortunately, most problems cannot be solved by mapping all operand lower bounds to the result lower bound and all operand upper bounds to the result upper bound. However, for most probabilistic evaluations of logic expressions, even these more difficult types of problems can be worked efficiently. Demonstrating this is a major aim of this paper.

1 The first member of the ordered pair is by convention the lower bound; the second is the upper bound. Also, the notation \( P(A) \) indicates that the uncertain parameter will have a particular value somewhere in the interval, and that value will be the same for every appearance of \( P(A) \).
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
Determining Result Bounds from the Bounds of the Operands

Determining the bounds of the probabilities of logic functions depends on finding a proper value for each of the constituent probabilities of the independent variables of the function. The disadvantage of a general Monte-Carlo-based search is that it is relatively inefficient (e.g., many values must be tested that are provably excluded [6]). Furthermore, when all of the known methods have been applied, there has apparently been no previous assured demonstration that exhaustive search of any remaining probabilistic variables is unnecessary. We have now provided a basis for a more efficient search involving only the bounds of the variables through a result that can be expressed as:

The contributors to the bounds of the probability of a logic function of independent variables can be obtained from a set of values including the bounds of the probabilities of each variable of the function.

Because of this result, only bounds need be considered in the following development. The first step in the development will be to establish a notation through which the subsequent results can be explained.

A Hueristic Notation

A notation for uncertain variables that helps clarify various aspects of the repeated variable problem is:

\[ x_i = X_i + \alpha_i \varepsilon_i \]  

where \( x_i = [x_i, x_u] \), \( X_i = \frac{x_i + x_u}{2} \), \( \alpha_i = \frac{x_u - x_i}{2} \), and \( \varepsilon_i = \pm 1 \) (negative for the lower bound; positive for the upper bound). Nonlinear operations (e.g., multiplication) produce products of the \( \varepsilon_i \) and multiple appearances of each \( \varepsilon_i \). As a result, the general notation will be \( \varepsilon_j = \pm 1 \) (whichever is appropriate for appearance \( j \) of the variable). This is important in the following reasoning for distinguishing between operations such as unconstrained mathematics, constrained mathematics, and Affine mathematics.

Example 1. Consider the probability range for an “Or” function of two independent interval variables. This example illustrates the use of the heuristic notation. It also illustrates why unconstrained mathematics, and another approach that has been proposed (Affine mathematics), cannot be depended on to produce the correct answer. Converted by the methodology of [1] into a disjoint set form and expressed in the above notation:

\[ y_1 = X_1 + X_2 - X_1X_2 + \alpha_1 \varepsilon_{11} + \alpha_2 \varepsilon_{21} - X_1 \alpha_2 \varepsilon_{22} - X_2 \alpha_1 \varepsilon_{12} - \alpha_1 \varepsilon_{12} \alpha_2 \varepsilon_{22} \]

\[ y_1 = X_1 + X_2 - X_1X_2 + \alpha_1 \varepsilon_{11} + \alpha_2 \varepsilon_{21} - X_1 \alpha_2 \varepsilon_{22} - X_2 \alpha_1 \varepsilon_{12} - \alpha_1 \varepsilon_{12} \alpha_2 \varepsilon_{22} \]

A proof appears in the Appendix.

---

\[ 2 \] A proof appears in the Appendix.
where the variables represent probabilities. For \( x_1 = [0.2, 0.4], x_2 = [0.7, 0.8], \)

\[
y_1 = 0.825 + 0.1e_{11} + 0.05e_{21} - 0.015e_{22} - 0.075e_{12} - 0.005e_{12}e_{22}
\]

For the unconstrained lower bound, \( e_{11} \) and \( e_{21} \) are negative, \( e_{12} \) and \( e_{22} \) are positive, and \( e_{12}e_{22} \) is identical to the product of \( e_{12} \) and \( e_{22} \), and therefore positive.

\[
y_{1ul} = 0.825 - 0.1 - 0.05 + 0.015 - 0.075 - 0.005 = 0.58
\]

For the unconstrained upper bound, \( e_{11} \) and \( e_{21} \) are positive, \( e_{12} \) and \( e_{22} \) are negative, and \( e_{12}e_{22} \) is identical to the product of \( e_{12} \) and \( e_{22} \), and therefore positive.

\[
y_{1uu} = 0.825 + 0.1 + 0.05 + 0.015 + 0.075 - 0.005 = 1.06
\]

Note that the “probability” greater than one is a byproduct of the error in unconstrained mathematics (not requiring \( e_{ij} \) and \( e_{ik} \) to have the same value). For the constrained lower bound, \( e_{11} = e_{12} \) and \( e_{21} = e_{22} \) are negative (the pairs must be identical by the constraint of being the same variable), and \( e_{12}e_{22} \) is therefore positive.

\[
y_{1cl} = 0.825 - 0.1 - 0.05 + 0.015 + 0.075 - 0.005 = 0.76
\]

For the constrained upper bound, \( e_{11} = e_{12} \) and \( e_{21} = e_{22} \) are positive (the pairs must be identical by the constraint of being the same variable), and \( e_{12}e_{22} \) is therefore positive.

\[
y_{1cu} = 0.825 + 0.1 + 0.05 - 0.015 - 0.075 - 0.005 = 0.88
\]

Another approach that has been proposed for these types of problems is the Affine mathematics approach, for which terms comprising products of each multiple of \( e \) terms are denoted with a separate \( e \) notation. Since the new notation is treated as independent of its constituents (which it is not), Affine mathematics gives only an approximation. The Affine approximation results in bounds that are always at least as wide as constrained bounds,\(^3\) but generally narrower than unconstrained bounds.

In the example, for the Affine lower bound, \( e_{11} = e_{12} \) and \( e_{21} = e_{22} \) are negative (the pairs must be identical by the constraint of being the same variable). However, \((e_{12}e_{22})\) is negative because it is treated as an independent entity.\(^4\) This leads to the Affine result:

\[
y_{1af} = 0.825 - 0.1 - 0.05 + 0.015 + 0.075 - 0.005 = 0.76
\]

\(^3\) This statement is proved in the Appendix.

\(^4\) Where Affine groupings are treated as if they were independent, the parentheses call attention to the approximation used.
For the Affine upper bound, \( e_{i1} = e_{i2} \) and \( e_{21} = e_{22} \) are positive. Also, \((e_{i2}e_{22})\) is positive because it is treated as an independent entity. This leads to the Affine result:

\[
y_{1ul} = 0.825 + 0.1 + 0.05 - 0.015 - 0.075 + 0.005 = 0.89
\]

This illustrates how Affine bounds can be wider than constrained bounds. Affine bounds are generally narrower than unconstrained bounds, but not always. As an illustration, consider the Affine solution to the "And" function of the two variables in Example 1.

\[
P(v_1 \cap v_2) = X_1X_2 + X_1\alpha_2e_{21} + X_2\alpha_1e_{11} + \alpha_1e_{11}\alpha_2e_{21}
\]

\[(4)\]

where \( x_i \) denotes \( P(v_i) \). For the Affine and the unconstrained lower bound, \( e_{i1} \) and \( e_{21} \) are negative. For the unconstrained lower bound, \( e_{12}e_{22} \) is positive (product of two negatives). However, for the Affine approach, \((e_{i2}e_{22})\) is negative, since it is treated as an independent entity. The unconstrained result is:

\[
P(v_1 \cap v_2)_{ul} = 0.225 - 0.015 - 0.075 + 0.005 = 0.13
\]

The Affine result is:

\[
P(v_1 \cap v_2)_{ul} = 0.225 - 0.015 - 0.075 - 0.005 = 0.12
\]

The upper bounds for the two approaches are identical (result is 0.32). This demonstrates that Affine bounds can be wider than unconstrained bounds.

**Example 2.** Consider a general solution for the Exclusive-Or function of two interval independent probability variables:

\[
y_2 = x_1 + x_2 - 2x_1x_2
\]

\[= X_1 + \alpha_1e_{11} + X_2 + \alpha_2e_{21} - 2X_1X_2 - 2X_1\alpha_2e_{22} - 2X_2\alpha_1e_{12} - 2\alpha_1\alpha_2e_{22}
\]

\[(5)\]

where the variables again represent probabilities. For \( x_1 = [0.2, 0.4], x_2 = [0.4, 0.8], \)

\[
y_2 = 0.54 + 0.1e_{11} + 0.2e_{21} - 0.12e_{22} - 0.12e_{12} - 0.04e_{12}e_{22}
\]

For the unconstrained lower bound: \( e_{11} = e_{21} \) are negative, and \( e_{22} = e_{12} \) are positive, so \( e_{12}e_{22} \) is positive.

\[
y_{2ul} = 0.54 - 0.1 - 0.2 - 0.12 - 0.12 - 0.04 = -0.04
\]

Note that the negative probability is another byproduct of unconstrained mathematics. For the unconstrained upper bound, \( e_{11} = e_{21} \) are positive, and \( e_{22} = e_{12} \) are negative, so \( e_{12}e_{22} \) is positive.
\( y_{2uu} = 0.54 + 0.1 + 0.2 + 0.12 + 0.12 - 0.04 = 1.04 \)

For the Affine lower bound, \( \epsilon_{11} = \epsilon_{12} \) are positive (since \( \alpha_1 - 2X_2\alpha_2 \) is negative), \( \epsilon_{21} = \epsilon_{22} \) are negative (since \( \alpha_2 - 2X_1\alpha_2 \) is positive), and \( (\epsilon_{12}, \epsilon_{22}) \) is positive.

\( y_{2al} = 0.54 + 0.1 - 0.2 + 0.12 - 0.12 - 0.04 = 0.40 \)

For the Affine upper bound, \( \epsilon_{11} = \epsilon_{12} \) are negative, \( \epsilon_{21} = \epsilon_{22} \) are positive, and \( (\epsilon_{12}, \epsilon_{22}) \) is negative.

\( y_{2au} = 0.54 - 0.1 + 0.2 - 0.12 + 0.12 + 0.04 = 0.68 \)

For the constrained lower bound, \( \epsilon_{11} = \epsilon_{12} \) are negative, \( \epsilon_{21} = \epsilon_{22} \) are negative, so \( \epsilon_{12}, \epsilon_{22} \) is positive.\(^5\)

\( y_{2cl} = 0.54 - 0.1 - 0.2 + 0.12 + 0.12 - 0.04 = 0.44 \)

For the constrained upper bound, \( \epsilon_{11} = \epsilon_{12} \) are negative, \( \epsilon_{21} = \epsilon_{22} \) are positive, so \( \epsilon_{12}, \epsilon_{22} \) is negative.

\( y_{2cu} = 0.54 - 0.1 + 0.2 - 0.12 + 0.12 + 0.04 = 0.68 \)

**Example 3.** Consider \( y_3 = v_1v_2 \cup \overline{v}_1v_3 \cup \overline{v}_1\overline{v}_2 \) (logic variables) expressed in terms of independent interval probabilities (\( x_i \) is the probability of \( v_i \)):

\[
y_3 = 1 + 2X_1X_2 - X_1 - X_2 + X_2X_3 - X_1X_2X_3 = 1 + 2X_1X_2 + 2X_1\alpha_2\epsilon_{21} + 2X_2\alpha_1\epsilon_{11} + 2a_1\epsilon_{11}\alpha_2\epsilon_{21} - X_1 - \alpha_1\epsilon_{12} - X_2 - \alpha_2\epsilon_{22} + X_2X_3 + X_2\alpha_3\epsilon_{31} + X_3\alpha_2\epsilon_{23} + \alpha_2\epsilon_{23}\alpha_3\epsilon_{31} - X_1X_2X_3 - X_1\alpha_2\epsilon_{24}\alpha_3\epsilon_{32} - X_2\alpha_1\epsilon_{13}\alpha_3\epsilon_{32} - X_3\alpha_1\epsilon_{13}\alpha_2\epsilon_{24} - X_2X_3\alpha_1\epsilon_{13} - X_1\alpha_2\epsilon_{24} - X_1X_2\alpha_3\epsilon_{32} - \alpha_1\epsilon_{13}\alpha_2\epsilon_{24}\alpha_3\epsilon_{32}
\]

(6)

where the variables again represent probabilities. For \( x_1 = [0.2, 0.4], x_2 = [0.7, 0.8], \) and \( x_3 = [0.6, 0.8], \)

\( y_3 = 0.7675 + 0.03\epsilon_{21} + 0.15\epsilon_{11} + 0.01\epsilon_{11}\epsilon_{21} - 0.1\epsilon_{12} - 0.05\epsilon_{22} + 0.075\epsilon_{31} + 0.035\epsilon_{23} + 0.005\epsilon_{23}\epsilon_{31} - 0.0015\epsilon_{24}\epsilon_{32} - 0.0075\epsilon_{13}\epsilon_{32} - 0.0035\epsilon_{13}\epsilon_{24} - 0.0525\epsilon_{13} - 0.0105\epsilon_{24} - 0.0225\epsilon_{32} - 0.0005\epsilon_{13}\epsilon_{24}\epsilon_{32} \)

For the unconstrained lower bound, \( \epsilon_{21}, \epsilon_{11}, \epsilon_{31}, \epsilon_{23} \) are negative, and \( \epsilon_{12}, \epsilon_{22}, \epsilon_{24}, \epsilon_{32}, \epsilon_{13} \) are positive, so \( \epsilon_{11}\epsilon_{21}, \epsilon_{23}\epsilon_{31}, \epsilon_{24}\epsilon_{32}, \epsilon_{13}\epsilon_{32}, \epsilon_{13}\epsilon_{24}, \) and \( \epsilon_{13}\epsilon_{24}\epsilon_{32} \) are positive.

\(^5\) The choices for the constrained approach are not always obvious. The necessary methodology follows subsequently.
\[ y_{3ul} = 0.7675 - 0.03 - 0.15 + 0.01 - 0.1 - 0.05 - 0.075 - 0.035 + 0.005 - 0.0015 - 0.0075 - 0.0035 - 0.0525 - 0.0105 - 0.0225 - 0.0005 = 0.244 \]

For the unconstrained upper bound, \( e_{21}, e_{11}, e_{31}, e_{23} \) are positive, and \( e_{12}, e_{22}, e_{24}, e_{32}, e_{13} \) are negative, so \( e_{11}e_{21}, e_{23}e_{31}, e_{24}e_{32}, e_{13}e_{32}, e_{13}e_{24} \) are positive and \( e_{13}e_{24}e_{32} \) is negative.

\[ y_{3uu} = 0.7675 + 0.03 + 0.15 + 0.01 + 0.1 + 0.05 + 0.075 + 0.035 + 0.005 - 0.0015 - 0.0075 - 0.0035 + 0.0525 + 0.0105 + 0.0225 + 0.0005 = 1.296 \]

For the Mine lower bound, \( e_{21} = e_{22} = e_{23} = e_{24} \) are negative, \( e_{11} = e_{12} = e_{13} \) are negative, \( e_{31} = e_{32} \) are negative, \( (e_{11}e_{21}) = (e_{13}e_{24}) \) is negative, \( (e_{23}e_{31}) = (e_{24}e_{32}) \) is positive, and \( (e_{13}e_{32}) \) and \( (e_{13}e_{24}e_{32}) \) are positive.

\[ y_{3al} = 0.7675 - 0.03 - 0.15 - 0.01 - 0.1 + 0.05 - 0.075 - 0.035 - 0.005 + 0.0015 + 0.0075 - 0.0035 - 0.0525 - 0.0105 + 0.0225 + 0.0005 = 0.69 \]

For the Mine upper bound, \( e_{21} = e_{22} = e_{23} = e_{24} \) are positive, \( e_{11} = e_{12} = e_{13} \) are negative, \( e_{31} = e_{32} \) are positive, \( (e_{11}e_{21}) = (e_{13}e_{24}) \) is positive, \( (e_{23}e_{31}) = (e_{24}e_{32}) \) is positive, and \( (e_{13}e_{32}) \) and \( (e_{13}e_{24}e_{32}) \) are positive.

\[ y_{3au} = 0.7675 + 0.03 - 0.15 - 0.01 + 0.1 - 0.05 + 0.075 + 0.035 + 0.005 - 0.0015 - 0.0075 + 0.0035 + 0.0525 - 0.0105 + 0.0225 - 0.0005 = 0.845 \]

For the constrained lower bound, \( e_{21} = e_{22} = e_{23} = e_{24} \) are positive, \( e_{11} = e_{12} = e_{13} \) are negative, \( e_{31} = e_{32} \) are negative, so \( e_{11}e_{21} \) and \( e_{13}e_{24} \) are negative, \( e_{23}e_{31}, e_{24}e_{32}, e_{13}e_{32}, e_{13}e_{24}e_{32} \) are positive.

\[ y_{3cl} = 0.7675 + 0.03 - 0.15 - 0.01 + 0.1 - 0.05 - 0.075 + 0.035 - 0.005 + 0.0015 - 0.0075 + 0.0035 + 0.0525 - 0.0105 + 0.0225 - 0.0005 = 0.704 \]

For the constrained upper bound, \( e_{21} = e_{22} = e_{23} = e_{24} \) are positive, \( e_{11} = e_{12} = e_{13} \) are negative, \( e_{31} = e_{32} \) are positive, so \( e_{11}e_{21} \) and \( e_{13}e_{24} \) are negative, \( e_{23}e_{31} \) and \( e_{24}e_{32} \) are positive, and \( e_{13}e_{32} \) and \( e_{13}e_{24}e_{32} \) are negative.

\[ y_{3cu} = 0.7675 + 0.03 - 0.15 - 0.01 + 0.1 - 0.05 + 0.075 + 0.035 + 0.005 - 0.0015 + 0.0075 + 0.0035 + 0.0525 - 0.0105 - 0.0225 + 0.0005 = 0.832 \]

As the examples demonstrate, choice of the signs of the \( e \) are individually deterministic for unconstrained mathematics, and deterministic through calculating coefficients of common variables in Affine mathematics. However, both methods cannot guarantee accurate bounds. The bounds for constrained mathematics are exact, but it is not obvious how to make the appropriate selection of the \( e \) to achieve the bounds. This problem will be approached in the following sections.
Example 4. As an introduction to the following concepts, consider the algebraic (and non-probabilistic) function:

$$y_4 = x_1x_2 + x_2x_3$$

(7)

where \(x_1 = [2,3]\), \(x_2 = [1,2]\), and \(x_3 = [-2,5]\). The unconstrained solution is \([-2,16]\), the Affine solution is \([-4, 16]\), and the constrained solution is \([0,16]\). Some insight into the subsequent subject of “unateness” can be facilitated by taking partial derivatives with respect to each variable:

$$\frac{\partial y_4}{\partial x_1} = x_2, \quad \frac{\partial y_4}{\partial x_2} = x_1 + x_3, \quad \frac{\partial y_4}{\partial x_3} = x_2$$

Since all three are non-negative, the lower bound of all operands can be used to determine the lower bound of the result, and the upper bound of all operands can be used to determine the upper bound of the result. This corresponds to the constrained mathematics solution of \([0,16]\). This “slope” information will be used to explain the effect of unateness in the next section.

Unate Variables

One important situation, which we address with our methodology, involves the probability evaluation of a Boolean function that is mate or has unate variables. Boolean functions are logical descriptions that can be applied to describe the outcomes of fault trees and event trees, among many other safety and reliability analysis applications.

Unateness [5] means that every variable of a Boolean function can be expressed such that each variable appears either complemented or uncomplemented, but both senses are not necessary. Any variable that meets this condition is called a unate variable (positive unate if uncomplemented, negative unate if complemented). Unateness is especially important in logical trees. An event tree or fault tree having only “Ands” and “Ors” such that each event affects the probabilistic outcome either positively everywhere it appears or negatively everywhere it appears can be represented by a unate logic expression. This tree condition is sufficient for unateness. Positive unate variables in a logic function map directly to positive partial derivatives in deriving the probability of the Boolean function, regardless of the values of the other variables. Conversely, negative unate variables map to negative partial derivatives, regardless of the values of the other variables.

A mathematically provable algorithm that accounts for constrained mathematics in an expression of probability for the logic function outcome is [6]:

For a positive unate variable, the lower bound of the result is a function of the lower bound of each appearance of the variable, and the upper bound of the result is a function of the upper bound of each appearance of the variable. For a negative unate variable, the lower bound of the result is a function of the upper bound of each appearance of the
variable, and the upper bound of the result is a function of the lower bound of each appearance of the variable.

This result naturally extends from independent variables to independent functions, as we will demonstrate in a subsequent example. Once unate variables have been processed, the solution for any non-unate variables can be traditional, but greatly simplified because of the removal of unate variables from the problem. Parsing for computer solution involves first determining the unate variables and their bounds, and then calculating the bounds for non-unate variables based on the restriction offered by the bounds of the unate variables. Finally, all variable bounds are combined to solve for the bounds of the result. The concepts in the above algorithm, the processing of non-unate variables, and the parsing order will be illustrated through examples.

Example 5. Consider the union of a positive unate variable and a negative unate variable:

\[ f_5 = x_1 \cup \overline{x}_2 \]  \hspace{1cm} (8)

A visual mode of representation is shown in Figures 1 and 2. On the left of Figure 1, the two variables are shown as orthogonal coordinates, and the four Boolean combinations of one and zero are shown as vertices on a two-dimensional "hypercube." In the center, the three combinations for which the function is satisfied are shown as black squares. On the right, a third dimension is added to show the function "ones" as black columns. In Figure 2, the probability of the variables is shown on two of the axes, and the probability of the logic function is shown in the third dimension. In contrast to the Boolean functions in Figure 1, the columns are point indicators of a surface, since the probabilities are continuous. The dashed lines show the pervasive monotonicity of the unate variables.

\[ f_5 = x_1 \cup \overline{x}_2 \]

Fig. 1. A Visual Mode of Representation for a Boolean Function

A visual mode of representation is shown in Figures 1 and 2. On the left of Figure 1, the two variables are shown as orthogonal coordinates, and the four Boolean combinations of one and zero are shown as vertices on a two-dimensional "hypercube." In the center, the three combinations for which the function is satisfied are shown as black squares. On the right, a third dimension is added to show the function "ones" as black columns. In Figure 2, the probability of the variables is shown on two of the axes, and the probability of the logic function is shown in the third dimension. In contrast to the Boolean functions in Figure 1, the columns are point indicators of a surface, since the probabilities are continuous. The dashed lines show the pervasive monotonicity of the unate variables.

---

\[^6\text{Any number of dimensions is theoretically possible.}\]
Example 6: Consider the logic function:

\[ f_6 = v_1 v_3 \cup v_2 v_4 \cup \overline{v}_2 \overline{v}_3 \]  

(9)

Here, \( v_1 \) and \( v_4 \) are positive unate, \( v_3 \) is negative unate, and \( v_2 \) is not unate. The probability expression is:

\[ y_6 = x_2 x_4 + \overline{x}_2 \overline{x}_3 + x_1 x_2 \overline{x}_3 \overline{x}_4 \]

where \( y_6 = P(f_6) \). The solution directly implements the bounds for \( x_1, x_3, \) and \( x_4 \) as:

\[ y_{6l} = x_2 x_{4l} + \overline{x}_2 \overline{x}_{3l} + x_1 x_2 \overline{x}_{3l} \overline{x}_{4u} \]  

and

\[ y_{6u} = x_2 x_{4u} + \overline{x}_2 \overline{x}_{3u} + x_1 x_2 \overline{x}_{3u} \overline{x}_{4l} \]

The solution for \( x_2 \) depends on the sign of the function of the other variables that are producted with \( x_2 \). Taking a partial derivative with respect to \( x_2 \):
\[ \frac{\partial y_5}{\partial x_2} = x_4 - x_3 + x_1 x_3 x_4 \]

Then if \( x_{4l} - x_{3u} + x_{1l} x_{3u} x_{4l} \geq 0 \), \( x_{2l} \) can be used in the computation for \( y_{4l} \). If \( x_{4l} - x_{3u} + x_{1l} x_{3u} x_{4l} \leq 0 \), \( x_{2u} \) can be used in the computation for \( y_{4l} \). If \( x_{4u} - x_{3l} + x_{1u} x_{3l} x_{4u} \geq 0 \), \( x_{2u} \) can be used in the computation for \( y_{4u} \). If \( x_{4u} - x_{3l} + x_{1u} x_{3l} x_{4u} \leq 0 \), \( x_{2l} \) can be used in the computation for \( y_{4u} \).

For \( x_1 = [0.1, 0.3] \), \( x_2 = [0.7, 0.9] \), \( x_3 = [0.4, 0.6] \), and \( x_4 = [0.6, 0.8] \), \( y_6 = [0.5512, 0.8124] \)

Many forms of logic expressions that do not meet any unateness criteria can be processed almost as efficiently as above. In order to demonstrate this, we will address the "exclusive-or" function (satisfied if an odd number of inputs are satisfied) and its inverse (satisfied if an even number of inputs are satisfied). These have in many respects characteristics completely opposite to unateness. For example, none of the partial derivatives of the variables reflect monotonicity.

**Example 7:** Consider the Exclusive-Or of two variables. Figure 3 portrays the two-dimensional Boolean function on the left and the three dimensional probability function on the right. The dashed lines on the probability surface indicate the non-monotonic nature through the partial derivatives of each variable as the other variables change. An effective procedure is to find the restricted area of the surface defined by the variable bounds and to test for monotonicity over that restricted area. Where the sense of the partial derivative changes over the restricted region, tabulation of the possibilities is still feasible.

As an illustration of the role of restrictions, consider the Exclusive-Or of two variables, where \( x_{1u} \leq 1/2 \) and \( x_{2l} \leq x_{2u} \). As the shaded area in Figure 4 indicates, the lower bound of the function is determined by the lower bound of both variables, and the upper bound of the function is determined by the lower bound of \( x_1 \) and the upper bound of \( x_2 \).
Figure 3. Exclusive-Or of Two Variables

Table 1. Exclusive-Or Bounds Rules

<table>
<thead>
<tr>
<th>Case</th>
<th>Condition</th>
<th>( y_i ) is a function of:</th>
<th>( y_u ) is a function of:</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>( w_u \leq 0.5, z_u \leq 0.5 )</td>
<td>( w_i ) and ( z_i )</td>
<td>( w_u ) and ( z_u )</td>
</tr>
<tr>
<td>b</td>
<td>( w_u \leq 0.5, z_i \geq 0.5 )</td>
<td>( w_u ) and ( z_i )</td>
<td>( w_u ) and ( z_u )</td>
</tr>
<tr>
<td>c</td>
<td>( w_i \geq 0.5, z_u \leq 0.5 )</td>
<td>( w_i ) and ( z_u )</td>
<td>( w_i ) and ( z_i )</td>
</tr>
<tr>
<td>d</td>
<td>( w_i \geq 0.5, z_i \geq 0.5 )</td>
<td>( w_i ) and ( z_u )</td>
<td>( w_i ) and ( z_i )</td>
</tr>
<tr>
<td>e</td>
<td>( w_u \leq 0.5, z_i \leq 0.5, ) ( z_u \geq 0.5 )</td>
<td>( w_i ) and ( z_i )</td>
<td>( w_i ) and ( z_u )</td>
</tr>
<tr>
<td>f</td>
<td>( z_u \leq 0.5, w_i \leq 0.5, ) ( w_u \geq 0.5 )</td>
<td>( w_i ) and ( z_i )</td>
<td>( w_u ) and ( z_i )</td>
</tr>
<tr>
<td>g</td>
<td>( w_i \geq 0.5, z_i \leq 0.5, ) ( z_u \geq 0.5 )</td>
<td>( w_i ) and ( z_u )</td>
<td>( w_i ) and ( z_i )</td>
</tr>
<tr>
<td>h</td>
<td>( z_i \geq 0.5, w_i \leq 0.5, ) ( w_u \geq 0.5 )</td>
<td>( w_i ) and ( z_u )</td>
<td>( w_i ) and ( z_i )</td>
</tr>
<tr>
<td>i</td>
<td>( w_i \leq 0.5, z_i \leq 0.5, ) ( w_u \geq 0.5, z_u \geq 0.5 )</td>
<td>( w_i ) and ( z_i ) ( \text{or } w_u ) and ( z_u )</td>
<td>( w_i ) and ( z_i ) ( \text{or } w_u ) and ( z_i )</td>
</tr>
</tbody>
</table>
Since these functions are linear and associative, they can be computed iteratively, which simplifies algorithmic implementation. A general algorithm will help illustrate these concepts.

For an exclusive-or of $n$ logic variables (or logic functions), consider two of the variables (or functions), $w$ and $z$, where $w$ and $z$ can each represent either some $x_i$ or $\bar{x}_i$. Nine possible cases can be specified, as illustrated in Table 1.

\begin{figure}[h]
\centering
\includegraphics{example_figure}
\caption{Example of the Role of Restrictions}
\end{figure}

**Example 8.** Consider the following logic function:

$$f_8 = v_1 \overline{v}_2 v_3 v_4 \cup v_1 \overline{v}_2 v_3 v_5 \cup v_1 \overline{v}_2 v_4 v_5 \cup v_1 v_2 v_3 v_4 \cup v_1 v_2 v_3 v_5 \cup v_1 v_2 \overline{v}_4 v_5$$

(10)

The probability is:

$$y_8 = x_1 \overline{x}_2 x_3 x_4 + x_1 \overline{x}_2 \overline{x}_4 x_5 + \overline{x}_1 x_2 x_3 x_4 + \overline{x}_1 x_2 \overline{x}_4 \overline{x}_5$$

, where $x_1 = [0.6, 0.7], x_2 = [0.4, 0.6], x_3 = [0.3, 0.5], x_4 = [0.5, 0.6], \text{ and } x_5 = [0.1, 0.3].$

When the logic function is simplified to:
\[ f_8 = v_1 \overline{v}_2 v_3 v_4 \cup v_1 \overline{v}_2 v_4 \overline{v}_5 \cup \overline{v}_1 v_2 v_3 v_4 \cup \overline{v}_1 v_2 \overline{v}_4 \overline{v}_5, \]  

(11)

it is apparent that \( v_3 \) is positive unate, and \( v_5 \) is negative unate. When the methodology of “compacting” [6] is applied to derive:

\[ f_8 = (v_1 \oplus v_2) v_3 v_4 + (v_1 \oplus v_2) \overline{v}_4 \overline{v}_5, \]  

(12)

it can be seen that \( v_1 \oplus v_2 \) serves the role of an independent unate subfunction. Using the information in the theorems, we can now determine that \( y_l \) is a function of \( x_{1u}, x_{2u}, x_{3l}, \) and \( x_{5u} \); and \( y_u \) is a function of \( x_{1u}, x_{2u}, x_{3u}, \) and \( x_{5l}. \) This leaves only the functionality of the bounds of \( x_4 \) to determine. Taking the partial derivative:

\[ \frac{\partial y_8}{\partial x_4} = (x_1 \oplus x_2)(x_3 - \overline{x}_5) \]

Since this derivative is everywhere negative, \( y_l \) is a function of \( x_{4u}, \) and \( y_u \) is a function of \( x_{4l}. \) The final result is: \( y_8 = [0.2116, 0.378]. \)

**Concluding Remarks**

Using the methodology developed in this paper, it is possible to accurately calculate safety and reliability probabilities with functions of repeated uncertain variables. The methodology is easily implemented in software, as we have partially done in our PHASER and COSMET fuzzy mathematics routines, and the results obtained are far superior to conventional software unconstrained operations, or even Affine approaches.

**Acknowledgments**

Several people facilitated these developments through detailed discussions on various aspects, and each reviewed at least part of the document. Of particular note are Prof. George Klir, State University of New York, Binghamton, Prof. Donald Wunsch, Texas Tech University, Prof. Timothy Ross, University of New Mexico, and Drs. Michael Bohn and Tim Trucano, Sandia National Laboratories. Partial funding was furnished by the U. S. Department of Energy through a Laboratory Directed Research and Development award.

**References**


Appendix: Mathematical Proofs

Theorem 1:

The contributors to the bounds of the probability of a logic function can be obtained from a set of values including the bounds of the probabilities of each independent variable of the function.

Proof:

First note that any logic function of independent variables can be expressed in “sum of minterms” form [7] (a logical sum of the logical product of all of the function variables, each appearing in either complemented or uncomplemented form). Since the minterms are by definition disjoint, the probability of the logic function can be expressed as the algebraic sum of the algebraic product of the probabilities of the function variables, each in the corresponding sense (complemented or uncomplemented) of the logical sum of minterms expression. For an n-variable logic function, \( f \), having \( m \) minterms, the sum of minterms expression can be denoted as:

\[
 f = \sum_{i=1}^{m} v_{i_1} v_{i_2} \ldots v_{i_n} \tag{A1}
\]

where the \( v_j (j=1,2,\ldots,n) \) can be either complemented or uncomplemented, the summation indicates logical sum, and the juxtaposition indicates logical product. The corresponding probability of the logic expression is:

\[
 y = \sum_{i=1}^{m} x_{i_1} x_{i_2} \ldots x_{i_n} \tag{A2}
\]

where the summation indicates algebraic sum and the juxtaposition indicates algebraic product.

Taking the partial derivative of Eqn. A2 with respect to \( x_j \):
\[ \frac{\partial y}{\partial x_j} = P(y \mid x_j = 1) - P(y \mid x_j = 0) \]  \tag{A3}

This derivative cannot be a function of \( x_j \), since \( x_j \) appears only once in each minterm. For any particular value of the other probabilistic variables, the partial derivative can be positive, negative, or zero. If it is positive, the lower bound of \( x_j \) determines the lower bound of the probability of \( f \), and the upper bound of \( x_j \) determines the upper bound of the probability of \( f \); if it is negative, the lower bound of \( x_j \) determines the upper bound of the probability of \( f \), and the upper bound of \( x_j \) determines the lower bound of the probability of \( f \).

If we exclude constant functions, then for any combination of values for the probabilities of the other variables, if a value other than the bounds gave a maximum or a minimum of \( y \), there would be a zero in the partial derivative and a nonzero value for \( x_j + \Delta x_j \) or \( x_j - \Delta x_j \) or both, which implies that the derivative would have to be a function of \( x_j \), contradicting Eqn. A3. Therefore, the bounds of \( y \) can be obtained from the bounds of the probabilities of the constituent variables of \( f \). For constant functions, the maximum is identical to the minimum and occurs everywhere, including at the bounds.

**Theorem 2:**

*The bounds of the probability of a logic function derived by Affine approximation can never be more narrow than the constrained mathematics bounds; specifically, the upper bound can never be lower, and the lower bound can never be higher.*

**Proof:**

Using the notation given in and following Eqn. 2 for representing the sign difference that discriminates lower bounds from upper, the solution for the \( \varepsilon_y \) product terms is exact under constrained mathematics, but an approximation under Affine mathematics. There are four possible situations for each product term:

1. The constrained solution for a lower bound requires a negative one for the exact result, in which case the Affine solution requires a negative one because of the lower bound. In this case the product term has the same value for both solutions.
2. The constrained solution for a lower bound requires a positive one for an exact result, in which case the Affine solution requires a negative one because of the lower bound. In this case the product term has a lower value for the Affine solution.
3. The constrained solution for an upper bound requires a negative one for the exact result, in which case the Affine solution requires a positive one because of the upper bound. In this case the product term has a higher value for the Affine solution.
4. The constrained solution for an upper bound requires a positive one for an exact result, in which case the Affine solution requires a positive one because of the higher bound. In this case the product term has the same value for both solutions.

As the product terms accumulate, each either contributes the same amount to both solutions or the Affine solution contributes to wider bounds than are necessary. Since the individual $\varepsilon_y$ terms and the constant terms contribute the same amount to both solutions, the theorem is proved.