QUANTITATIVE SAFETY ANALYSIS

J.W. CROACH AND L.M. ARNETT

Savannah River Laboratory
Aiken, South Carolina
LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:
A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.
QUANTITATIVE SAFETY ANALYSIS

by

J. W. Croach, Director
Technical Division

and

L. M. Arnett
Computer Applications Division
Computer Sciences Section

April 1970

E. I. DU PONT DE NEMOURS & COMPANY
SAVANNAH RIVER LABORATORY
AIKEN, S. C. 29801

CONTRACT AT(07-2)-1 WITH THE
UNITED STATES ATOMIC ENERGY COMMISSION
ABSTRACT

An analytical method is developed to assess the safety of an operation or a mission. A computer program, PAR, is written to implement the quantitative features of this method. The analysis can incorporate judgment factors and can assess the results of accepting each of several alternatives.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>5</td>
</tr>
<tr>
<td>DISCUSSION</td>
<td>6</td>
</tr>
<tr>
<td>Concept</td>
<td>6</td>
</tr>
<tr>
<td>Mathematical Model</td>
<td>8</td>
</tr>
<tr>
<td>Computational Strategy</td>
<td>13</td>
</tr>
<tr>
<td>Implicit Bivariant Interdependence</td>
<td>13</td>
</tr>
<tr>
<td>Zero Entries</td>
<td>15</td>
</tr>
<tr>
<td>Segmentation</td>
<td>16</td>
</tr>
<tr>
<td>Relative Consequences and Risk</td>
<td>21</td>
</tr>
<tr>
<td>PAR Computer Program</td>
<td>23</td>
</tr>
<tr>
<td>Data Specification</td>
<td>25</td>
</tr>
<tr>
<td>Logic</td>
<td>27</td>
</tr>
<tr>
<td>Input</td>
<td>31</td>
</tr>
<tr>
<td>Sample Problem</td>
<td>33</td>
</tr>
</tbody>
</table>

## APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Card Input Data</td>
<td>41</td>
</tr>
<tr>
<td>B</td>
<td>Sample Problem with 25 Bivariates</td>
<td>42</td>
</tr>
<tr>
<td>C</td>
<td>NCV Array</td>
<td>42</td>
</tr>
<tr>
<td>D</td>
<td>KSTR, KHEX, and PROST Arrays</td>
<td>43</td>
</tr>
<tr>
<td>E</td>
<td>Summary of Results</td>
<td>44</td>
</tr>
<tr>
<td>F</td>
<td>FORTRAN Listing</td>
<td>45</td>
</tr>
</tbody>
</table>

- 3 -
INTRODUCTION

Two important barriers to the application of quantitative probabilities in safety analysis are:

- Lack of convenient analytical methods of adequate scope and flexibility
- Difficulty of assembling pertinent input data

These two problems are related. If an analytical approach were available that had great flexibility in accepting and handling a wide range of input data, the task of providing the data might be easier. This is especially important when some quantitative input data must reflect primarily judgment factors. It then becomes essential to have an analytical framework in which the consequences of such judgment may be quickly and conveniently depicted.

This report describes an analytical approach and a computer program that incorporates broad scope and flexibility. The program Probabilistic Analysis of Risk (PAR) is available for possible application. Further improvements in the capability of PAR are believed to be possible if computing time proves to be a limitation.

SUMMARY

An analytical approach is developed to assess the safety of an operation, campaign, or mission. The information of possible concern at the end of the campaign or after a time period of the operation is contained in yes or no answers to a set of appropriate propositions or questions. Each combination of yes and no answers corresponds to a unique state or condition of the system. If there are n questions in the set, then there are \(2^n\) possible states. The objective of the analysis is to find for each question the unconditional probability that its answer is yes. Whereas, these unconditional probabilities may not be known within orders of magnitude on the basis of either experience or judgment, their dependence upon the answers to other questions may be much more reliably known. This situation is a direct consequence of the increase in the reliability of an answer with an increase in the amount of information available to support the answer.
The analytical method consists of three steps:

1. Formulation of an appropriate set of questions whose answers will provide the desired information.

2. Assignment of conditional probabilities to each question that its answer is yes, given each possible state of the set of remaining questions upon which the question of concern is dependent.

3. Calculation for each question of the unconditional probabilities that its answer is yes.

Although the judgment exercised in the formulation of the input will obviously influence the results of the analysis, the nature of the judgment is explicitly expressed in quantitative measures. Possible disagreements among people concerned with a specific problem should be capable of reconciliation because of the quantitative detail of the analysis.

The computer program PAR is written to accept the type of input described in Step 2 and to calculate the unconditional probabilities of the answer to each question. Also calculated is the risk associated with questions that have undesirable consequences when their answer is yes. The risk is calculated as the product of a relative undesirability factor and the unconditional probability of a yes answer. An important feature of the computer program is the logical procedures it utilizes to minimize the computing time for large problems.

DISCUSSION

CONCEPT

A statistical or probabilistic evaluation of an operation, program, or mission is desired. If a steady-state operation is to be evaluated, an appropriately short period of time is selected for statistical characterization: perhaps one day of reactor operation. An appropriately short period is one in which multiple occurrence of events to be characterized may be neglected. For instance, a major power failure has a certain probability of occurring in one day; the likelihood of two different, independent failures in one day may ordinarily be safely neglected. For convenience, the general term "campaign" will describe the specific mission, program, or interval of operating time that is to be characterized.
The campaign is considered to be specified by information that describes the intended course of the campaign and anticipated possible deviations from it. This information includes designation of equipment, instrumentation, protective systems, backup devices, personnel, procedures, and administrative controls, as well as specified objectives and plans. From the point of view of hazards analysis, attention is primarily focused on assessment of probabilities that various undesirable departures from the nominal campaign will occur.

The analytical procedure is based on construction of a set of propositions such that a particular outcome of the campaign may be adequately characterized by indicating for each proposition whether it is true or false. The method resembles the old game of "Twenty Questions" in which only yes or no answers are permitted. In some respects, the method is a generalized fault tree approach. The portrayal of the spectrum of all campaign outcomes may be quantitatively represented by the interdependence of the propositions and by the probabilities that each proposition will be true (or false). Interdependence is important. For instance, if one of the propositions happened to be (for a hypothetical day of reactor operation) "A reactor incident occurred"; and if this statement is false for a particular campaign outcome, then another proposition, "A major power failure occurred," can hardly be true for the same outcome. A key point in the analysis is that the occurrence probabilities of many events of interest may be given conveniently in terms of their dependence upon other related events. A computer program based on an appropriate mathematical model may conveniently accept such interdependent data and systematically calculate probabilities that each proposition will be true or false.

If many possible, undesirable consequences of widely differing magnitudes and probabilities must be considered, the analysis should provide a method for combining probability and magnitude of a particular consequence in a single quantity that may be treated as a risk. We assume that the risk associated with a particular undesirable consequence C is measured by the product:

\[ \text{Risk} = P_C R_C \]

where \( P_C \) is the probability of the occurrence of C in the campaign, and \( R_C \) is a relative measure of the undesirability of the consequence C. We assume that risks defined in this way may be compared one to another and that they may be added to obtain a meaningful combined risk.
The concept of risk as a product of probability and relative undesirability is of secondary importance in the analysis. Estimation of consequence probabilities is the primary objective. Assignment of quantitative relative undesirabilities of consequences may serve a useful purpose in reflecting the relative weight judged to be appropriate for diverse consequences.

The analytical method would be most effectively applied in an iterative feedback procedure as follows: A set of propositions, interdependent probabilities, and undesirable consequence factors are analyzed to obtain unconditional event (or proposition) probabilities and risks. The results of the analysis are then reviewed to determine if the desired features of the campaign have been adequately described by the set of propositions and if the unconditional probabilities and risks are in accordance with judgment and experience. The problem is then reformulated and the analysis is repeated. The iterative process continues until the objectives of the analysis have been satisfactorily met. The value of such an analysis does not lie in a few isolated quantitative probabilities and risks, but rather in the systematic portrayal of the hazards structure as viewed by the analyst. The analysis should afford a more quantitative medium of communication for exchange of views and judgments on hazards analysis. Areas of agreement and disagreement are more easily identified among people engaged in analysis, review, and approval of hazardous programs. Hopefully, a dissenter should be able to put his finger on specific points of the analysis with which he disagrees or believes to be inadequate.

More details of the method will become evident in the discussions of the mathematical model and the computer program in subsequent sections. (The reader may profitably scan the Sample Problem, page 33, at this point to get a qualitative feel for the formulation of a problem.)

MATHEMATICAL MODEL

Everything of concern about the outcome of the campaign is assumed to be definable by yes and no answers to a finite set of questions. For example, the answers to five questions might be sufficient in a particular instance to provide all the desired information. In this case, $2^5 = 32$ different outcomes are represented by all combinations of yes and no answers to the five questions. If yes is represented by 1 and no by 0, then each of the 32 combinations corresponds to a binary number, which can be used to identify the outcome. This set of outcomes forms a mutually exclusive and exhaustive set because one, and only one, outcome must result for each campaign or experiment.
Each question or proposition about the campaign is considered to be represented by a binary variable which may assume only one of the two values 1 (yes = true) and 0 (no = false). These propositions are the basic elements of the model and are called bivariates. The bivariates are designated $X_1, X_2, \ldots, X_n$, and the ordered set is designated $S_n$. The set of all possible outcomes or all possible combinations of 1 and 0 for the bivariates is designated $Z_n$. Therefore, the set $Z_n$ always contains $2^n$ elements. To continue the example of five questions, the particular outcome

$$X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 1, \text{ and } X_5 = 0$$

corresponds to the binary number 10110 or the decimal number 22.

This particular element of the $Z$ set is designated $Z_{5,22}$ and is one of the 32 elements in the complete set. The 32 elements in $Z_5$ are considered to have the standard order $Z_{5,0}, Z_{5,1}, Z_{5,2}, \ldots, Z_{5,31}$.

The set of bivariates (propositions) $S_n$ is selected for a given campaign so that all outcomes of possible interest are included in the set $Z_n$. One would expect to select bivariates in such a way that, in principle, a yes or no could be assigned to each defining proposition on the basis of available post-campaign records. For instance, after the fact, it can be said for any day's operation of a reactor whether or not there was a power failure, a scram, a power surge accident, etc.

A campaign described by a bivariant set $S_n$, viewed as a statistical entity, can be completely characterized if the probabilities of the occurrence of each outcome or element of the set $Z_n$ are specified. For example, when five different coins are tossed, all statistical information about the experiment is contained in the statement that all 32 outcomes are equally likely or that the probability of each outcome is $1/32$. All other statistical questions that one might ask, such as:

What is the probability that the third coin is heads?

What is the probability that there are at least three heads among the five coins?

can be answered from the knowledge of the probabilities of the individual outcomes. These latter questions deal with events rather than single outcomes. Each of these events is defined by a subset of outcomes, and the probability of an event is the sum of the probabilities of the outcomes defining the event. In roulette, the set of mutually exclusive and exhaustive outcomes consists of 00, 0, 1, 2, \ldots, 36 while odd, even, red, black, etc., are events, each of which is defined by a subset of the outcomes.
The probability that \( Z_{nj} \) is the outcome of the campaign is defined to be \( Q_{nj} \). Since one of the \( Z_{nj} \) must occur,

\[
\sum_{j=0}^{2^n-1} Q_{nj} = 1 \tag{1}
\]

If the \( Q_{nj} \) are known for all \( j \), then the other statistical quantities related to the sets \( S_n \) and \( Z_n \) can be calculated. Specifically, the probability that the campaign outcomes will yield a yes for the \( k \)th bivariable \( (X_k = 1) \) is the sum of all \( Q_{nj} \) for the \( Z_{nj} \) in which \( X_k = 1 \). In symbols,

\[
P \{X_k = 1\} = W_k = \sum_{j=0}^{2^n-1} X_{kj} Q_{nj} \tag{2}
\]

where \( X_{kj} \) is either 1 or 0 in accordance with whether \( X_k = 1 \) or 0 in \( Z_{nj} \). The probability of any other event of interest in the campaign can be calculated by summing the \( Q_{nj} \) for the subset of outcomes in \( Z_n \) that define the event.

The basic relation from which all of the desired information is calculated from available input data depends on one of the fundamental laws of probability, viz., the multiplication theorem. The multiplication theorem may be expressed as

\[
P \{A \times B\} = P \{B|A\} P \{A\} \tag{3}
\]

In words, the theorem states that the probability that both events \( A \) and \( B \) will occur equals the product of the probabilities of events (1) and (2), where (1) is \( B \) occurs given that \( A \) has occurred, and (2) is \( A \) occurs. For example, what is the probability that two cards drawn at random from a bridge deck will both be hearts? The theorem states that

\[
P \{H_2 \times H_1\} = P \{H_2|H_1\} P \{H_1\}
\]

\( P \{H_1\} \) is just the probability that the first card is a heart and is clearly equal to \( 13/52 = 1/4 \). The quantity \( P \{H_2/H_1\} \) is the probability that the second card is a heart given that the first card is a heart. When the first card is a heart, the remaining 51 cards include 12 hearts so that \( P \{H_2|H_1\} = 12/51 \) and finally
\[ P \{ H_2 \times H_1 \} = \frac{12}{51} \cdot \frac{1}{4} = \frac{1}{17} \]

Expressions of the form \( P \{ B|A \} \) are known as conditional probabilities and imply a possible dependence of the event \( B \) upon knowledge concerning the outcome of other or previous events \( A \). In the example given above, the probabilities of drawing cards depend upon what has already been drawn. There are instances, of course, in which the events are independent, as, for example, when a coin is tossed repeatedly.

The multiplication theorem can be extended to more than two events, as follows:

\[ P \{ A \times B \times C \} = P \{ C|A \times B \} P \{ B|A \} P \{ A \} \quad (4) \]

Equation (4) permits the computation of \( Q_{nj} \) for set \( Z_n \) in terms of the interdependence of the bivariates in the set \( S_n \).

The ordered sequence of \( S \) and \( Z \) sets is

\[ S_1, S_2, S_3, \ldots, S_n \text{ and } Z_1, Z_2, Z_3, \ldots, Z_n \]

Each set in the two sequences is generated by adding another bivariate, say \( X_m \), to the \( m-1 \) bivariates previously defined as \( S_{m-1} \). If the dependence of the probability of \( X_m \) on the elements of \( Z_{m-1} \) is known, and if all \( Q_{m-1,j} \) for the set \( Z_{m-1} \) are known, then all of the \( Q_{nj} \) can be calculated. The multiplication theorem is applied as follows:

\[ P \{ (X_m = 1) \times Z_{m-1,j} \} = P \{ (X_m = 1)|Z_{m-1,j} \} P \{ Z_{m-1,j} \} \quad (5) \]

The event \( X_m \) and \( Z_{m-1,j} \) is a member of the \( Z_m \) set, and the probability of the event is some \( Q_{mi} \). The relation between the \( i \) and \( j \) will be resolved. The quantity \( P \{ Z_{m-1,j} \} \) has been defined as \( Q_{m-1,j} \). The notation for the quantity \( P \{ (X_m = 1)|Z_{m-1,j} \} \) is shortened to \( P_{mj} \). For \( X_m = 0 \), the conditional probability is \( 1 - P_{mj} \). Therefore, equation (5) is now written as

\[ Q_{mi} = P_{mj} \cdot Q_{m-1,j} \quad (6) \]
Induction shows that when $X_m = 1$ is added to $Z_{m-1,j}$, the new outcome is $Z_{m,2j+1}$. Likewise, when $X_m = 0$ is added to $Z_{m-1,j}$, the new outcome is $Z_{m,2j}$. Thus, the recurrence relations are:

$$Q_{m,2i+1} = P_{mi} \cdot Q_{m-1,i}$$

(7)

$$Q_{m,2i} = (1 - P_{mi})Q_{m-1,i}$$

All the $Q_{nj}$ and thus all the probabilities of events of interest can be calculated when the $Q_{11}$ for the initial proposition and all the $P_{mi}$ for subsequent propositions are specified. The $Q_{11}$ is just the probability that the initial proposition is true. Obviously, $Q_{10} = 1 - Q_{11}$. The conditional probabilities $P_{mi}$ are the quantities that are specified when the problem is defined and thus they constitute the input to the calculational problem.

In theory, the analytical process is now outlined. A bivariable set $S_m$ is defined, and the sequence order of the $X_i$ is selected so that one can specify the dependence of any $X_m$ on all the $X_i$ appearing earlier in the sequence, i.e., $i < m$. One might proceed in this fashion theoretically to define and analyze campaigns that require any number of bivariables.

The quantitative specification of the problem in terms of conditional probabilities has the advantage that these are the quantities that can be stated most reliably either on the basis of past observations or merely on judgment. For example, the probability that a given person will have a serious accident in an automobile during the next ten minutes is strongly dependent upon answers to such questions as:

- Is the driver intoxicated?
- Is the automobile speed greater than 80 mph?
- Is the next mile of road crooked and mountainous?

If the answers to all three questions were yes, then the probability of a serious accident would usually be judged as high. Obviously, any information about the conditions under which the truth of a proposition is being judged will assist in making the most appropriate quantitative assignment. The answers to these three questions may in turn be considered dependent upon still other propositions and may go all the way back to the question,

- Is the person in an automobile?
for which the unconditional probability is stated. There may, of course, be more than one question for which an unconditional probability is given.

**COMPUTATIONAL STRATEGY**

The practical feasibility of the analytical approach for systems of large \(n\) depends on the effectiveness of methods to cope with the potentially large dimensionality of the model. To illustrate the potential size, a set of 100 bivariates \((n = 100)\) may be regarded as defining a problem in 100-dimensional space with a possible manifold of \(2^{100}\) or about \(10^{33}\) elements. The conditional probabilities \(P_{99,i}\) that are required to compute the set \(Q_{100}\) for the set \(Z_{100}\) must be defined for \(2^{99}\) elements. To list all the input data for such a problem is obviously impractical, much less to solve it by brute-force application of equation (7). On the other hand, an analysis of a system that requires several hundred bivariates is not an unreasonable objective.

In general, the concepts for dealing with potentially large dimensionality all take into account features that may be expected in a practical representation of a problem. Three useful concepts are:

1. Implicit bivariative interdependence

2. Exploitation of a high proportion of zero entries in the input data

3. Segmentation of a large bivariative set into loosely coupled subsets

Each of these concepts is discussed in the following sections.

**Implicit Bivariative Interdependence**

When a set \(S_n\) of \(n\) bivariates has been defined, the dependence of an arbitrary \(X_m\) in the set on the preceding bivariates is completely given by equation (5), which is repeated here:

\[
p_{mi} = P\{X_m = 1|Z_{m-1,i}\} = \frac{P\{X_m = 1 \times Z_{m-1,i}\}}{P\{Z_{m-1,k}\}} = f_m(Z_{m-1,i})
\]

(5)
In practice, the conditional probabilities \( P_{mkj} \) may frequently be expected to depend on a subset \( K \) of \( k \) bivariates contained within the set \( m-1 \), \( k < m-1 \). All possible states of the \( k \) bivariates of \( K \) are reflected by a subset \( Z_k \) within \( Z_{m-1} \). Equation (5) may then appear as

\[
P_{mkj} = F_{mk}(Z_{kj})
\]

(8)

where the specific \( X_i \) that comprise the set \( K \) have been identified. The first subscript \( m \) on the \( P \) and \( F \) is retained to indicate that the \( m \)th bivariates is being considered, and the additional subscripts denote that the domain of the function has been shortened from \( Z_{m-1} \) to \( Z_k \). \( Z_{kj} \) is the binary number defined by the ordered set of the \( k \) different \( X_i \), when the \( X \) values of 0 and 1 are assigned. Thus, the subset \( (k = 3), X_1, X_5, X_3 \), of the set \( (m-1 = 10), X_1, X_2, \ldots, X_{10} \) will generate a sequence of binary numbers \( Z_{31} = 0, Z_{32} = 1, \ldots, Z_{38} = 111 \). In this example, there are eight values for \( P_{mkj} \) corresponding to the eight \( Z_{kj} \).

The implicit definition of all the \( P_{mi} \) in equation (5) is equivalent to an agreement to derive these values from Equation (8): For a given set \( Z_{m-1}, i \) (which is itself a binary number), all bits are omitted that do not correspond to \( X_i \) in the subset \( K \). The remaining bits in their original order define a binary number \( Z_{kj} \) for which a \( P_{mkj} \) is defined as input data. Thus, there is one \( P_{mkj} \) for each of the elements in the set \( Z_{kj} \). The subset \( K \) may be null, e.g., \( k = 0 \). If so, only one \( P_{mkj} \) is defined; viz., \( P_{m0} \).

The functional dependence given in equation (8) implies that the \( k \) bivariates upon which the probability of occurrence of \( X_m \) depends have been explicitly listed. This explicit list is referred to as the argument list of \( X_m \). All indices of the argument list must be less than \( m \). The entire function of equation (8) is specified by a set of \( 2^k \) \( P_i \) that must be provided as input data to calculate the probability of the occurrence of \( X_m \). This set is referred to as the probability function set, or simply function set, for \( X_m \). To repeat the previous definition, each of the \( P_i \) is the conditional probability that the event \( X_m \) will occur given the occurrence of the corresponding set of events upon which \( X_m \) is dependent.
The efficiency of this concept (really a convention) is apparent from the previous example. The function set for the bivariant $X_{11}$ is represented by

$$P_{11,3} = F_{11,3} (X_1, X_5, X_9)$$

This is a set of eight values, but it serves to define all $2^{13} = 1024$ values in the full function set for $X_{11}$ implied by equation (5).

Zero Entries

Even though all possible bivariates may not appear in the argument lists for the respective $X_i$, it may be expected that sizeable argument lists must be handled. A list of twenty to thirty does not seem out of reason. However, an argument list of thirty implies a function set of over a billion $P_i$. Fortunately, a practical representation of such a data set will quite likely contain a very high proportion of zeros. The high density of zeros may stem from two sources: (1) an accurate formulation of the problem may truly yield a high zero density, and (2) knowledge of the problem by the analyst may permit him to assign zeros to many probability entries. The latter situation arises because the analyst knows that the effect on the results of interest will be negligible.

The following circumstances are examples of conditions that will produce high zero densities. In the first instance, the argument list for some bivariant $X_m$ may include ten earlier bivariates so that the possible function set comprises 1024 entries. It might be known that $X_m$ can be true only when at least nine out of the ten bivariates in the argument list have the value 1 (yess). In this case, there would be only eleven nonzero entries in the function set of $X_m$. These correspond to binary numbers of ten bits with no more than one zero bit. All other probability entries are zero. The zero density in this case is about 99%.

In the second instance, the analyst might know, for example, that $X_1 = 1$, $X_5 = 1$, and $X_9 = 1$ in the argument list of $X_{11}$ would certainly cause $X_{11}$ to equal 1 or

$$P = F(1,1,1) = 1.0$$

However, he might also know that $X_5$ and $X_9$ are so interrelated that $X_9$ cannot possibly occur if $X_5$ does. The probability entry of 1 is correct, but academic, because the corresponding simultaneous occurrence of $X_5$ and $X_9$ is impossible. The analyst then
knows that his result will be unchanged if he considers
\[ P = F(1,1,1) = 0. \]
In practice, the analyst may choose zero
entries for probabilities in a function set whenever he judges
the contributions of such entries to be negligible. Bivariates
may be defined and selected in such a way that the results of
the analysis provide a cross-check on a priori judgments of this
sort.

High zero density in the input data may be exploited both
in the input medium and during computation by explicitly handling
only nonzero elements. Each nonzero probability in the input
probability function sets may be associated with an identifier
derived from the binary number corresponding to it in the set of
all states of the argument list. Specification of this identifier
for a particular bivariates permits retrieval of the matching
probability; failure to find an identifier in its appropriate
array means that the corresponding probability is zero. Compu-
tational advantages are more apparent in the subsequent discussion
of the computer program.

Segmentation

The computing time for a problem with \( n \) bivariates may be
roughly proportional to the number of possible states that must
be considered; i.e., \( 2^n \), or approximately \( 2^n \). If such a problem
could somehow be reduced to the solution of two independent
problems, each comprising \( n/2 \) bivariates, the total number of
states in the two combined problems is given approximately as
\( 2 \cdot 2^{n/2} \). In comparison with the original problem, this is a
reduction by a factor of \( 2^{n/2-1} \). If \( n \) is a large number, say
100, the reduction factor is indeed impressive — on the order
of \( 10^{14} \)! If the two problems are not completely independent
of each other so that some additional computation is required
to account for interactions between them, it is apparent that
a substantial increase in computing time for each of the separated
problems would still be more than compensated by the ability to
segment. In the example with \( n = 100 \), a reduction in computing
speed by a factor of 100 (to account for interactions) would still
yield an improvement factor of about \( 10^{14} \) for the segmented
problem compared to the original.

The foregoing paragraph is intended to illustrate qualitatively
the potential computational improvement that may be anticipated
for segmentation of a large problem into smaller ones. In
practical problems, the expected high zero density discussed earlier
may cause computing time for a set of \( n \) to vary more like \( a^n \), where
\( a \) is a number smaller than 2 (but greater than 1). Even so, it
may be expected that considerable benefits in computing time will
accrue if segmentation is exploited.

- 16 -
The optimum exploitation of segmentation could get to be a complicated problem in itself. The optimum would depend upon the character of the typical input data to be processed. Qualitatively, the optimum approach would be one in which the decrease in computing time enabled by enhanced capability to segment is just offset by the increased computational complexity (and time) required to account for interactions. The interactions that are treated in the present version of PAR are comparatively simple. If computing time is of real concern in practical problems, the treatment of more complex interactions may be justified.

To become more specific about interaction, assume a set $S_n$ of $n$ bivariates has been defined, and the argument list for each bivariative is specified. Any bivariative $X_i$ may be considered to generate a chain that forms a subset of $S_n$. The chain based on $X_i$ is defined as that subset of $S_n$ that includes $X_i$ and the chains of all bivariates that contain $X_i$ in their argument lists. The definition of "chain" is not circular as it might appear. The stepwise generation of the chain from $X_i$ is as follows: $X_i$ is the first element; all $X_i$'s that contain $X_i$ in their argument lists are identified and added to the chain. Next, those $X_i$ are identified that contain within their argument lists any $X$ already within the chain; those $X_i$ so identified are added to the chain. The process is repeated until all bivariates within the set $S_n$ have been resolved as either in or not in the chain of $X_i$. From the definition of a chain, it is apparent that there must be at least one bivariative in a chain, and there may be as many as $n$ bivariates.

It is now possible to deal more precisely with interaction. Two bivariates $X_i$ and $X_j$ are said to interact if, and only if, the chains based on $X_i$ and $X_j$ have at least one common element. Or conversely, $X_i$ and $X_j$ may be said to have no interaction if the intersection of their chains is null. The underlying thought behind the interaction of bivariates is this: If there is no interaction between $X_i$ and $X_j$, then it will not be necessary in the ultimate solution of the problem to account for all four possible states of the pair $X_iX_j$ (00, 01, 10, 11) as they influence either the probabilities of $X_i$, $X_j$, or any other bivariative that depends upon them indirectly; $X_i$ and $X_j$ may therefore be separated with impunity.

A calculable subset within $S_n$ is defined as a subset that satisfies the following requirement: Each bivariative in the subset contains within its argument list (if any) only bivariates that are also members of the same calculable subset. Since the indices of bivariates in any argument list are required to be less than the index of the bivariative itself, at least one bivariative of a calculable subset must have a null argument list. In other words, the absolute, unconditional probability must be given for at least one bivariative in a calculable subset.

- 17 -
Examples of calculable subsets are (each bivariant is represented by its index):

**Example 1**

<table>
<thead>
<tr>
<th>Bivariant</th>
<th>Argument List</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Null</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1,2</td>
</tr>
<tr>
<td>4</td>
<td>1,3</td>
</tr>
<tr>
<td>10</td>
<td>2,3,4</td>
</tr>
</tbody>
</table>

**Example 2**

<table>
<thead>
<tr>
<th>Bivariant</th>
<th>Argument List</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Null</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>

The following subset is not calculable:

<table>
<thead>
<tr>
<th>Bivariant</th>
<th>Argument List</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>10,11</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>12,13,14</td>
</tr>
</tbody>
</table>

In the last example, the subset is not calculable because of the presence of bivariates 10 and 11 in the argument list of 12. No. 10 and 11 are not members of the subset. This subset may become calculable as a result of a preceding computation of a calculable subset that includes 10 and 11, so that the absolute probability of No. 12 is determined.

Now assume that a calculable subset \( A \) within \( S_\pi \) has been somehow selected and that the interaction of \( A \) with the remainder of \( S_\pi \) is to be considered. (The intent is to determine whether the subset \( A \) may be usefully segmented from \( S_\pi \) for computational purposes.) Define another subset \( B \) in \( S_\pi \) that will be called the set of external bivariates associated with \( A \). An external bivariate (by definition): (1) is not contained in \( A \) and (2) contains within its argument list at least one bivariate that belongs to \( A \).
If the set of external bivariates, \( E \), is null, then \( A \) may obviously be segmented from the rest of \( S_\n \) and computed as a separate group. The subset \( E \) could contain no bivariates only if all chains originating within the set \( A \) also are contained in \( A \); in which case \( A \) is completely independent of the remainder of \( S_\n \).

If \( E \) is not null, there is some interaction between \( A \) and the remainder of \( S_\n \). (Notationally not \( A \) in \( S_\n \) is denoted \( \bar{A} \).) A measure of the degree of interaction between \( A \) and \( \bar{A} \) can be gained from the intersection of all chains based in \( E \), i.e., the intersection of all chains based on the external bivariates of \( A \); define this intersection as set \( B \). A case of particular interest is where \( B \) is null. In this case, it can readily be shown that no pairs of chains emanating from bivariates within \( A \) have an intersection anywhere in \( \bar{A} \) except in \( E \). If \( B \) is null, none of the bivariates in \( E \) interact with each other. The interaction of \( A \) with the individual bivariates of \( E \) can be resolved when the computation for \( A \) is executed. After the computation of subset \( A \), the final result is:

1. Unconditional probabilities for all bivariates in \( A \) are determined.

2. The remaining subset \( \bar{A} \) is free of any residual interactions among bivariates of \( A \).

If \( B \) is not null, e.g., some bivariates of \( E \) interact, computation of the calculable subset will not permit resolution of all interactions between \( A \) and \( \bar{A} \). The bivariates of \( A \) in argument lists of interacting members of \( E \) may not be disposed of once and for all and must be treated subsequently in one or more calculable subsets. If any members of \( E \) are isolated (e.g., do not interact with the other members of \( E \)), it is possible to achieve a reduction of their argument lists through computation of \( A \). Also, if there are members of \( A \) that do not appear in any argument list of interacting members of \( E \), those members may be resolved; e.g., it will not be necessary to treat them again.

Define an actionable subset as a calculable subset that has at least one isolated external bivariative (member of \( E \)) or that has one or more resolvable members.
The segmentation algorithm of PAR is based on the following:

1. Search the unresolved bivariates to find the actionable subset with fewest members.

2. Compute subset; obtain unconditional probabilities for all subset members (if any) that may be resolved. Reduce argument lists of isolated members of E. Go to 1.

This algorithm is based on the assumption that any progress toward solution that may be achieved in small subset calculations is preferred to even much greater apparent progress through computation of larger subsets. Undoubtedly more effective algorithms could be devised, but they would require more elaborate methods for analyzing and utilizing the special features of particular problems.

Return now to the reduction of argument lists of isolated bivariates of E at the time the computation of A is carried out. Suppose \( X_k \) is a particular isolated bivariative of E. Let the function set for \( X_k \) be given as

\[
P_i = F_k(Y_A, Y_{\bar{A}}) \tag{9}
\]

where \( Y_A \) and \( Y_{\bar{A}} \) denote the two subsets of \( X_k \) argument list that contain respectively the bivariates in \( A \) and in \( \bar{A} \). The basic procedure is to utilize the detailed state probabilities computed for \( A \) (and therefore for \( Y_A \)) to derive a new, reduced probability function set for \( X_k \) in the form,

\[
P_i = F'_k(Y_A) \tag{10}
\]

If \( Y_A \) and \( Y_{\bar{A}} \) contain \( p \) and \( q \) bivariates respectively, then the index "i" in equation (9) ranges over at most \( 2^{(p+q)} \) elements, while in (10) it ranges over at most \( 2^q \). The reduction of (9) and (10) can be expressed as

\[
P_j = \sum_{i=0}^{2^p-1} Q_i F_k(Y_A, Y_{\bar{A}}) = F'_k(Y_{\bar{A}}) \tag{11}
\]
The notation here is a bit mixed; \( Y_{A_i} \) and \( \hat{Y}_A \) denote specific state combinations of bivariates in \( Y_A \) and \( Y_A \) that may be identified with binary state numbers as was done for equation (5). The \( Q_i \) are individual probabilities for the occurrence of the possible states of \( Y_A \). For a specified \( i \)th state of \( Y_A \), \( Q_i \) may be computed as the sum of all those individual state probabilities for \( A \) for which the specified \( i \)th state of \( Y_A \) occurs. Since the computation of the subset \( A \) requires the calculation of all nonzero state probabilities in \( A \) anyhow, the \( Q_i \) may be accumulated as a byproduct of the subset computation.

In the preceding discussion, interaction among bivariates is considered to exist if there is a potential for interaction on the basis of the prescribed argument lists. It is possible that the specific values assigned to the function sets of potentially interacting bivariates are such that no actual interaction occurs. Or if interaction does numerically occur, it may have a negligible effect on the results. It is conceivable that methods for testing interaction against some predefined tolerance could be incorporated in the segmentation algorithm, but these possibilities have not been investigated.

RELATIVE CONSEQUENCES AND RISK

In the first section, the concept of relative consequence values was introduced.

As before, in the general scheme of the model, a set \( S_n \) of \( n \) bivariates has been defined. The \( 2^n \) possible states of these bivariates can be represented by the set \( Z_n \) of all \( n \)-digit binary numbers. A particular member of \( Z_n \), say \( Z_{nj} \), corresponds to a particular state (or outcome) in which the status of each bivariate is specified as either zero or one. The probability of the \( Z_{nj} \)th state is \( Q_{nj} \). Equation (2) gives the absolute or unconditional probability that the \( k \)th bivariate will have a yes (\( X_k = 1 \)) for the campaign.

\[
W_k = \sum_{j=1}^{2^n-1} X_{kj} Q_{nj}
\]

(2)

If a consequence value \( C_j \) is assigned to each possible state \( Z_{nj} \), then in the usual fashion, the expected consequence of the campaign, \( \bar{R} \), is defined as
\[
\bar{R} = \sum_{j=0}^{2^n-1} c_j q_{nj}
\] (12)

Conceivably, the consequence function that defines \( c_j \) could be any function set with domain \( Z_n \) and a range of real numbers. If consequence values were to reflect both desirable and undesirable campaign outcomes, both positive and negative \( c_j \) might be encountered; here, however, the \( c_j \) are taken to represent only undesirable consequences and, for convenience, all \( c_j \) are taken as positive, real numbers. A particular functional form that permits considerable simplification is to assume that each bivariable \( x_i \) has a consequence value \( r_i \) associated with it such that the consequence value \( c_j \) for any state \( z_{nj} \) is equal to the sum of the bivariable consequences \( r_i \) for which \( x_i = 1 \) in \( z_{nj} \); or

\[
c_j = \sum_{i=1}^{n} x_{ij} r_i
\] (13)

In equation (13) as before, \( x_{ij} \) is assigned the value \((0 \text{ or } 1)\) that \( x_i \) has in the \( z_{nj} \)th state. Combining equation (12) and (13) gives

\[
\bar{R} = \sum_{j=0}^{2^n-1} x_{1j} q_{nj}
\]

but

\[
\sum_{j=0}^{2^n-1} x_{1j} q_{nj} = w_1
\]

therefore

\[
\bar{R} = \sum_{i=1}^{n} w_i r_i
\] (14)

Another way to state the foregoing functional dependence is to say that the occurrence of \( x_i = 1 \) contributes a consequence \( r_i \) to the overall consequence, regardless of the states and consequence contributions of other bivariates. This might be considered as a postulated independence of consequences. This approach seems to correspond well with intuitive notions. Further, a method is readily available to account for any interactions among consequences for which there is a basis in judgment. For
example, suppose a set $S_n$ is defined and that one wishes to assign a consequence for the simultaneous occurrence of $X_j = 1$ and $X_k = 1$ that is greater than $R_j + R_k$. It is easy to add another biviable $X_e$ to the set $S_n$ so that $X_e = 1$ only if $X_j = X_k = 1$. A consequence $R_e$ may be assigned to $X_e$ explicitly to account for consequence interactions of $X_j$ and $X_k$. Thus the occurrence of $X_j = 1$ or $X_k = 1$ (but not both) is represented by consequence $R_j$ or $R_k$. Simultaneous occurrence of $X_j$ and $X_k$ is represented by consequence $R_j + R_k + R_e$.

The units of consequence values must be consistent within any analysis and among problems in which it is desired to compare risks. In the PAR program, the units are arbitrary and the values are identified as Relative Undesirable Estimates (RUE), which must be provided as input.

**PAR COMPUTER PROGRAM**

The PAR computer program (Probabilistic Analysis of Risks) is an experimental embodiment of the analytical approach previously described. It is at a stage where it might be usefully employed in actual hazards analysis.

PAR is written in FORTRAN IV and will now accept up to 200 bivariates. We assume that the analyst groups his bivariates data in a way that recognizes the segmentation logic built into the code. Injudicious grouping could result in the definition of a logically acceptable problem that would require prohibitive computing time. The principal question of applicability of the code concerns the amount of computing time required for practical problems of interest.

The discussion of PAR is divided into sections that deal with (1) data specification, (2) logic, and (3) input. The output features are shown in the following section on Sample Problem. The overall logical flow within the program is shown as follows:
Logical Flow for PAR Program

Start

Read Data

Write Data

Options

Select Subset

Subset Defined?

No

Yes

Initiate Subset Computation

Compute Risks and Cumulative Risks

Write Summary

Stop

Yes

No

Compute Next Nonzero Subset State Probability

Assign Probability Increments

Last Subset State?

Yes

Reduce External Bivariates and Reset Arrays
Data Specification

The principal working arrays and mnemonics are:

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEV(200)</td>
<td>KEV(I) is the number of bivariates in the argument list of the Ith bivariates.</td>
</tr>
<tr>
<td>NCV(200,32)</td>
<td>NCV(I,J) identifies the Jth bivariates in the argument list of I. J &lt; I.</td>
</tr>
<tr>
<td>KSTR(200)</td>
<td>KSTR(I) is the address of the initial entry for function set data in KHEX and PRØST for the Ith bivariates.</td>
</tr>
<tr>
<td>KDAT(200)</td>
<td>KDAT(I) is the number of data pairs entered in KHEX and PRØST for the Ith bivariates, beginning at KSTR(I).</td>
</tr>
<tr>
<td>KHEX(3000)</td>
<td>State-identifier for argument list of I; number is formed by assigning binary bits from left to right as 1 or 0 in accordance with the state of J in the ascending argument list NCV(I,J).</td>
</tr>
<tr>
<td>PRØST(3000)</td>
<td>PRØST(K) is the probability that the pertinent bivariates is 1 (yes) if the state of its argument list is given by KHEX(K).</td>
</tr>
<tr>
<td>RUE(200)</td>
<td>RUE(I) is the relative undesirability estimate for the occurrence of the one-state of the Ith bivariates.</td>
</tr>
<tr>
<td>NET(200)</td>
<td>Scratch-pad arrays for developing structural relationships.</td>
</tr>
<tr>
<td>NIP(200)</td>
<td></td>
</tr>
<tr>
<td>NUP(200)</td>
<td></td>
</tr>
<tr>
<td>NGRP(200)</td>
<td>NGRP(J) is the bivariates index of the Jth member of the current actionable subset.</td>
</tr>
<tr>
<td>NW(200)</td>
<td>NW(M) = 0 or 1 indicates bivariates status as respectively unresolved or resolved.</td>
</tr>
<tr>
<td>NB(200)</td>
<td>During subset computation, the specification of NB(1), NB(2), ..., NB(NCT) as (each) either 1 or 0 defines the current state of the actionable subset. See Logic.</td>
</tr>
<tr>
<td>Mnemonic</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PMUL(200)</td>
<td>In computing the nonzero probability PRØB of a specific subset state, PMUL(I) is the probability that the Ith bivariable has a one-state.</td>
</tr>
<tr>
<td>PRCM(3000)</td>
<td>PRCM(J) accumulates the probability factor that is applied to PRØST(J). This takes place in the reduction of external variables. See Logic.</td>
</tr>
<tr>
<td>ABSPRØ(200)</td>
<td>ABSPRØ(I) is the absolute or unconditional probability that I has a one-state (yes) for the campaign.</td>
</tr>
<tr>
<td>RISK(200)</td>
<td>At output, RISK(I) = ABSPRØ(I) * RUE(I)</td>
</tr>
<tr>
<td>CURISK(200)</td>
<td>At output, CURISK(I) is the accumulated sum of RISK(J) for J &lt; I.</td>
</tr>
<tr>
<td>NFAM(200)</td>
<td>NFAM(I) = 1 denotes a family generator. See Logic.</td>
</tr>
<tr>
<td>NCT</td>
<td>Number of bivariables in the current actionable subset.</td>
</tr>
<tr>
<td>NZØ</td>
<td>NZØ = (10^{-30}); in testing for zero in entries of the PRØST array, any number less than (10^{-30}) is considered as zero.</td>
</tr>
<tr>
<td>NVAR</td>
<td>Bivariable index on input card.</td>
</tr>
<tr>
<td>NTØT</td>
<td>Total number of bivariables defined at input.</td>
</tr>
<tr>
<td>NFØRM</td>
<td>Specifies form of input data for a particular bivariable. See Input.</td>
</tr>
<tr>
<td>KØPT</td>
<td>Options for listing contents of working arrays. See Input.</td>
</tr>
<tr>
<td>NMAX</td>
<td>Maximum permissible subset size. Unless intentionally overridden on input, NMAX is 25.</td>
</tr>
</tbody>
</table>

The integer function LGLAND(I,J) appears several places in PAR. The value returned by this function is the integer constructed by a logical AND of the corresponding binary bits of I and J. For example:

\[
I = 25 \text{ (binary 11001)} \\
J = 21 \text{ (binary 10101)} \\
LGLAND(I,J) = 17 \text{ (binary 10001)}
\]
Subroutine FAM(J,NET) generates the calculable subset in which J is the subset member of highest index. Members of the calculable subset are designated in NET(200) as NET(1) = 1. FAM has access to the arrays NCV and KEV through labeled COMMON /SAVE/.

Logic

The discussion of logic is intended only to give a qualitative picture of the coding with respect to Selection of Actionable Subsets and Subset Computation, including the Reduction of External Bivariates.

Wherever symbols are employed that are not code mnemonics, but which might be mistaken as such, they are introduced in quotes.

Selection of Actionable Subsets

At least one bivariative in a set defined for a problem has the property that it does not appear in the argument list of any other bivariative. (The bivariative of highest index in the set always has this property.) Such bivariatives are called generators. Each generator is associated with a family, which is the calculable subset derived from the generator. Two different families may have bivariatives in common — and, in fact — may differ only in their generators. The significance of a family is that any pair of bivariatives interact if they belong to the same family.

It is possible to identify the isolated (if any) external bivariatives of any calculable subset by determining if a candidate is or is not a member of some family that contains at least one other external bivariative. A calculable subset is considered to be actionable if at least one external bivariative is isolated. (It is also possible for a subset to contain a single resolvable member with no external bivariatives and thereby to qualify as actionable.)

The next actionable subset to be computed is found by determining that actionable subset which has the smallest number of members. Ties are broken by selecting the actionable subset of lowest maximum index. If the selected actionable subset contains more than the specified NMAX number of members, the program terminates with a note to that effect and gives a partial summary of results up to that point.
Subset Computation

Assume that A is the current actionable subset and that it contains NCT members. The bivariates that comprise A are ordered by increasing index K and identified in NGRP as NGRP(1), ..., NGRP(K), ..., NGRP(NCT). The possible states of A may be represented as the set of m-bit binary numbers Z. In the code, the role of the binary number Z is assigned to the array NB(J). NB(1) represents the left-most bit and NB(NCT) is the right-most; integers 0 and 1 are the only values assigned.

The subset computation proceeds by systematically finding all nonzero state probabilities of A. Initially all NB are zeroed, corresponding to the initial binary number Z = 0. The correspondence between Z and NB is as follows:

<table>
<thead>
<tr>
<th>Z</th>
<th>NB(NCT)</th>
<th>NB(NCT-1)</th>
<th>NB(NCT-2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The possible states of A are explored to find the nonzero state probabilities by proceeding systematically through the sequence of $Z_i$ from 0 to $2^{n-1}$ (n = NCT). The specification of a particular Z (overall state of the subset) also determines the state of each bivariable argument list so that a unique probability for each bivariable can be assigned from the input data. The probability of each Z state is the product of the individual bivariable probabilities. Not all Z states must be considered explicitly; since this fact is the basis for exploiting high zero density, further amplification is in order.

For convenience, assume the potentially nonzero states are ordered in sequence, $Z_1$, $Z_2$, ..., $Z_i$ .... Suppose $Z_i$ has just been considered and it is desired to determine the next Z that must be considered in detail. If the bits of Z are ordered 1, 2, 3, ..., from the right, suppose the first nonzero bit of $Z_i$ is the kth bit. Then there are two rules that govern the selection of $Z_{i+1}$, depending upon whether the bivariable corresponding to the kth bit does or does not have a zero probability in state $Z_i$. If the probability is not zero, then the next state that must be considered is $Z_{i+1} = Z_i + 1$. If the probability is zero, the next state that must be considered is

$$Z_{i+1} = Z_i + 2^{k-1}$$
The reason for this is that the state of the argument list for the bivariant represented by the kth bit is specified by bits of Z of the left of the kth bit; if this state probability is zero, it will remain zero for all possible Z's that have the same bits as \( Z_i \) in the kth and leftward positions. The next potentially nonzero Z in the sequence is therefore obtained from \( Z_i \) by arithmetically inserting 1 in the kth bit of \( Z_i \); this is the equivalent of adding \( 2^{k-1} \) to \( Z_i \).

In the extreme limiting case where all input probabilities are zero, the number of possible states that must be considered in a subset A of m bivariates is m instead of \( 2^m \).

When a nonzero state probability for A is found and computed as PR\( \theta \), the increment is accumulated in the array ABSPR\( \theta \) (I) for all resolvable bivariates in subset A; i.e., the current value of PR\( \theta \) is added to ABSPR\( \theta \) (I) if I represents a resolvable bivariative with a one-state in the current Z state for A. The accumulated value of ABSPR\( \theta \) (I), when the subset computation is complete, is the unconditional probability that the ith bivariative has a value of 1 for the campaign.

Reduction of External Bivariates

The isolated external bivariates in E are prepared for reduction as the subset computation for A proceeds. The current value of PR\( \theta \) is appropriately accumulated in PRCM(J), where J matches the position of an entry in the array PR\( \theta \) ST(J) that stores the input probability function sets for all bivariates. The reducible members of E are found by searching NIP(I) to find those elements tagged with integer 4.

The reduction step may be conveniently described symbolically as follows: Consider the probability function set for a particular reducible member of E. Suppose it has n bivariates in its bivariative list; let m be the number of these bivariates in A, and k be the number in \( \bar{A} \). Make the following definitions with respect to the bivariates of the argument list:

\[
\begin{align*}
Z_T &= \text{n-bit binary state identifier for the argument list} \\
Z_A &= \text{m-bit binary state identifier for the bivariates in A} \\
Z_{\bar{A}} &= \text{k-bit binary state identifier for bivariates in } \bar{A}
\end{align*}
\]
ZA is the binary number obtained by suppressing or ignoring the m bits in ZT related to A; ZA is the binary number similarly derived from ZT by ignoring the bits related to A. Thus for any specific ZT there corresponds a unique pair of ZA and ZA. For example, if the argument list is X_1, X_2, X_3, X_5, X_{10} and X_1, X_3, and X_{5} are in A and if ZT = 11010,

then

\[ Z_A = 1\underline{0}010 = 101 \]

and

\[ Z_A = 1\underline{1}0010 = 10 \]

The probability function set (for the particular reducible external bivariate being considered) consists of a list of specific ZT values (stored in KHEX), each with a corresponding probability (stored in PR0ST). When a nonzero probability increment PR0B has been computed for a specific state of A, say Z₁, the appropriate bits of Z₁ permit the specification of a ZA which may be compared with the ZA's of each function set pair (as derived from ZT). Where a match is found for some J, PR0B is accumulated in the corresponding PRCM(J) position. When the subset computation is finished, the original function set probabilities in the PR0ST array are now modified in the reduction step as follows:

\[ PR0ST(J) = PRCM(J) * PR0ST(J) \]

The binary identifier, ZT, previously stored in KHEX(J) is replaced by ZA. At this point, there may be several entries in the function set that have the same KHEX values (ZA). The function set is scanned to find such multiple entries and to consolidate them in a single entry; the corresponding values of PR0ST are accumulated as a total in a matching entry. The working arrays NCV, KEV, and KDAT are modified to reflect the changes brought about by reduction of external variables.

In the event all bivariates in the argument list of a reducible external bivariate J are included in the subset A, the reduction process eliminates all bivariates from the argument list of J, and the single remaining entry in the PR0ST array for J represents the absolute probability of that bivariable. The corresponding KEV(J) will become zero (corresponds to a null argument list).
Card input data are arranged as follows:

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Card Columns</th>
<th>Mnemonic</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-72</td>
<td>-</td>
<td>72H</td>
<td>Problem title.</td>
</tr>
<tr>
<td>2</td>
<td>1-5</td>
<td>NTØT</td>
<td>I5</td>
<td>Total number of bivariates.</td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td>KØPT</td>
<td>I5</td>
<td>Option specifying listing of input data.</td>
</tr>
<tr>
<td>11-15</td>
<td>NMAX</td>
<td></td>
<td>I5</td>
<td>Maximum permissible size of subset (If blank, program sets NMAX = 25).</td>
</tr>
<tr>
<td>16-20</td>
<td>KHD</td>
<td></td>
<td>I5</td>
<td>Specifies I/O Format to KHEX (see Card Type 5).</td>
</tr>
<tr>
<td>3</td>
<td>1-5</td>
<td>NVAR</td>
<td>I5</td>
<td>Index number of bivariante (must follow serial order I = 1, 2, 3, ... NTØT).</td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td>NPØRM</td>
<td>I5</td>
<td>Specifies card type that follows current Card 3 or its continuation. Type 4 for NPØRM = 0; Type 5 for NPØRM = 1.</td>
</tr>
<tr>
<td>11-25</td>
<td>RUE(I)</td>
<td></td>
<td>E15.5</td>
<td>Relative undesirability of Ith bivariante.</td>
</tr>
<tr>
<td>26-30</td>
<td>KE</td>
<td></td>
<td>I5</td>
<td>Number of bivariates in argument list of I.</td>
</tr>
<tr>
<td>31-35</td>
<td>KA</td>
<td></td>
<td>I5</td>
<td>Number of data pairs for I (needed only for NPØRM = 1).</td>
</tr>
<tr>
<td>36-80</td>
<td>NCV(I,J)</td>
<td></td>
<td>11I5</td>
<td>Bivariante indices in argument list of I, monotonically increasing order.</td>
</tr>
<tr>
<td>3'</td>
<td>1-80</td>
<td>NCV(I,J)</td>
<td>16I5</td>
<td>Continuation of 3 if needed.</td>
</tr>
<tr>
<td>3&quot;</td>
<td>1-25</td>
<td>NCV(I,J)</td>
<td>5I5</td>
<td>Continuation of 3' if needed.</td>
</tr>
<tr>
<td>4</td>
<td>1-80</td>
<td>PRØST</td>
<td>16E5.2</td>
<td>KE**2 probability entries are expected. Sequence of entries must track binary states of argument list. Use as many Type 4 cards as required. Type 4 must follow 3 if NPØRM = 0 on 3.</td>
</tr>
<tr>
<td>Card Type</td>
<td>Card Columns</td>
<td>Mnemonic</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>---------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>5</td>
<td>1-10</td>
<td>KHEX</td>
<td>I10</td>
<td>Type 5 must follow 3 if NFORM = 1. KHEX is read as I10 if KHD = 0; KHEX is read as Z10 if KHD = 1. KHEX and PRØST, for any pair, represent a single argument list state and a corresponding probability. Five pairs per card in 5(I10,E5.2) or 5(Z10,E5.2) Format for as many cards as required to make KA pairs (see Type 3).</td>
</tr>
<tr>
<td></td>
<td>11-15</td>
<td>PRØST</td>
<td>E5.2</td>
<td>Repeat Types 3/4 or 3/5 for each I until all NTØT bivariates are listed.</td>
</tr>
<tr>
<td>etc.</td>
<td>thru 75</td>
<td></td>
<td></td>
<td>Note: Use of Z format for entering KHEX data is especially convenient since each number derived from an argument list state is basically a binary number to begin with so that conversion to hexadecimal is easier than construction of a decimal integer.</td>
</tr>
</tbody>
</table>
Input Limitations

Maximum number of bivariates
(Working arrays could be redimensioned to accommodate many more bivariates if desired.)

Maximum number of bivariates in any argument list

Maximum number of total pairs of entries in KHEX and PRØST

Options

Optional listing of initial data in the working arrays is controlled by the value of KØPT on Card 2. Available choices are:

<table>
<thead>
<tr>
<th>KØPT</th>
<th>Output List</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>For each bivariable index I, KDAT(I) and KEV(I) are listed</td>
</tr>
<tr>
<td>1</td>
<td>As above plus listing of NCV array</td>
</tr>
<tr>
<td>2</td>
<td>Same as 0 plus listing of KSTR and KHEX</td>
</tr>
<tr>
<td>3</td>
<td>Combine 0, 1, and 2</td>
</tr>
<tr>
<td>6</td>
<td>Combine 0, 2 plus PRØST array</td>
</tr>
<tr>
<td>7</td>
<td>Combine 0, 1, 2 plus PRØST</td>
</tr>
</tbody>
</table>

SAMPLE PROBLEM

A short sample problem is described in detail to illustrate the application of PAR. The problem comprises a set of 25 bivariates that relate to the operation of an imaginary test reactor. In the example, two types of hazard are treated. The first type has to do with the possibility of losing control of the reactor while it is being brought to power—a startup accident. The second type relates to a power surge accident. Obviously a complete hazards analysis for an actual reactor would require consideration of many more potential accidents in much greater depth. The propositions that represent the bivariates are intended to be illustrative of the kind that might actually be used. However, it must be recognized that each proposition in practice would require careful definition to avoid ambiguity in the classification of possible events. The input probability and relative undesirability data in the sample problem are pure fiction.
The 25 propositions that form the bivariable set are given in order. Each proposition is intended to apply to a randomly selected day. Thus the campaign is one day. In each case, the argument list is given and the nonzero probabilities for designated states of the argument list are shown. The state identifiers (Arg State) are given as hexadecimal numbers.

<table>
<thead>
<tr>
<th>No.</th>
<th>Bivariable Proposition</th>
<th>Arg State</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The reactor operated.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: Null</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Departure from standard operation was required.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>There was an operating error.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 1 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>There was a serious operating error.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>A startup accident occurred (SUA).</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 1 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SUA. Period meter &quot;A&quot; was bypassed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SUA. Period meter &quot;B&quot; was bypassed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>SUA. Flux monitor &quot;C&quot; was bypassed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>SUA. Flux monitor &quot;D&quot; was bypassed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>Bivariant Proposition</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SUA. Flux monitor &quot;E&quot; was bypassed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5 8 9</td>
<td>Arg State</td>
<td>Prob</td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1 0</td>
<td>4</td>
<td>.10</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>5</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 0</td>
<td>6</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>7</td>
<td>.001</td>
</tr>
<tr>
<td>11</td>
<td>SUA. Flux trip level set too high.</td>
<td>Arg State</td>
<td>Prob</td>
</tr>
<tr>
<td></td>
<td>Arg List: 5</td>
<td>1</td>
<td>.10</td>
</tr>
<tr>
<td></td>
<td>RUE = 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>3</td>
<td>.10</td>
</tr>
<tr>
<td>12</td>
<td>SUA. Reactivity addition rate exceeded x % k per second.</td>
<td>Arg State</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 4 5</td>
<td>1</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>RUE = 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>5</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>6</td>
<td>.02</td>
</tr>
<tr>
<td></td>
<td>1 1 3</td>
<td>7</td>
<td>1.0</td>
</tr>
<tr>
<td>13</td>
<td>SUA. Period scram failed</td>
<td>Arg State</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5 6 7</td>
<td>4</td>
<td>.001</td>
</tr>
<tr>
<td></td>
<td>RUE = 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>5</td>
<td>.002</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>6</td>
<td>.02</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>7</td>
<td>1.0</td>
</tr>
<tr>
<td>14</td>
<td>SUA. Flux level scram failed.</td>
<td>Arg State</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 5 8 9 10</td>
<td>8</td>
<td>.0001</td>
</tr>
<tr>
<td></td>
<td>RUE = 0 1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>9</td>
<td>.0001</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>A</td>
<td>.0001</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>B</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>C</td>
<td>.0001</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>D</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>E</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>F</td>
<td>1.0</td>
</tr>
<tr>
<td>15</td>
<td>SUA. Some fuel was melted.</td>
<td>Arg State</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 11 12 13 14</td>
<td>2</td>
<td>.001</td>
</tr>
<tr>
<td></td>
<td>RUE = 1 1 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>3</td>
<td>.01</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>6</td>
<td>.10</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>7</td>
<td>.50</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>E</td>
<td>.20</td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>F</td>
<td>1.0</td>
</tr>
<tr>
<td>16</td>
<td>SUA. More than 10% of fuel melted.</td>
<td>Arg State</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arg List: 12 15</td>
<td>1</td>
<td>.05</td>
</tr>
<tr>
<td></td>
<td>RUE = 10 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 1 1</td>
<td>3</td>
<td>.50</td>
</tr>
<tr>
<td>No.</td>
<td>Bivariant Proposition</td>
<td>Arg List</td>
<td>RUE</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------------------------------</td>
<td>----------</td>
<td>-----</td>
</tr>
<tr>
<td>17</td>
<td>SUA. More than 50% of fuel melted.</td>
<td>12 16</td>
<td>500</td>
</tr>
<tr>
<td>18</td>
<td>SUA. Reactor was disrupted.</td>
<td>17</td>
<td>500</td>
</tr>
<tr>
<td>19</td>
<td>A power surge accident occurred (PSA).</td>
<td>1 3 4</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 1 1</td>
<td>7</td>
</tr>
<tr>
<td>20</td>
<td>PSA. Temperature monitor trip level set too high.</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>PSA. Temperature operating limit in use was incorrectly high.</td>
<td>19 20</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 1</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>PSA. Unwanted control rod motion occurred.</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>PSA. Temperature monitor scram failed.</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>PSA. Some fuel melted.</td>
<td>4 20 21 22 23</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0 0 1 1</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 1 1 0 1</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 1 0 0 0</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 1 1 0</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0 0 1 0</td>
<td>2</td>
</tr>
<tr>
<td>25</td>
<td>PSA. More than 10% of fuel melted.</td>
<td>4 24</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 1</td>
<td>3</td>
</tr>
</tbody>
</table>
Note: The inclusion of SUA or PSA in the proposition is to be interpreted to mean that the designated accident occurred as well as whatever is otherwise asserted by the proposition.

Appendix A shows the card input data for the problem; B, C, D show the output listing of the data obtained with K0PT = 7 as an option; and E summarizes the results.

If the results are taken at face value for the sample problem, the risk of melting fuel via a power surge accident far outweighs the risk from a startup accident. The largest single contribution to the total risk is the risk of No. 25: that over 10% of the fuel might be melted in a power surge accident. The probability of such an accident on a randomly selected day is computed as 2.7 x 10^-6. Presumably if the operating characteristics, administrative policies, and instrumentation remained the same for an indefinite period so that the input data continued to apply, one could expect such an accident to occur with a mean-time-between-failure of about 1000 years.

Some points in connection with the sample problem are discussed in the following:

Selection of Bivariable Propositions

It should be evident that there is a great deal of freedom in the selection of propositions. Obviously there is no single, unique way to go about it. The important factors to keep in mind are:

1. Pertinence of propositions to the objective of the analysis
2. Nature of the available data
3. Structure of the bivariable set to facilitate computation

Assignment of Relative Undesirability Estimates (RUE)

In the example, RUE values were taken as zero for all bivariables except those that involve actual damage to the fuel or to the reactor. Where nonzero RUE values were assigned, the magnitudes were intentionally selected to reflect possible influences of subjectivity in judging consequences. For instance, RUE for melting more than 10% of the fuel in a startup accident (16) is 10; the RUE value for melting more than 50% of the fuel in a startup accident is 500. The gross nonlinearity is meant to reflect a possible judgment that the larger accident is a great deal more serious than is reflected by the amount of fuel
melted. Also for 10% or greater melting, the RUE for the power surge accident is ten times that for the startup accident; this might reflect the presence of a fission product inventory in the former accident.

Interpretation of Argument List Data

Consider a particular proposition: say bivariable (13), which states in full, "A startup accident occurred and the period scram failed." The function set for the argument list indicates conditional probabilities contingent upon the state of three bivariables of lower index:

5. A startup accident occurred (SUA).
6. SUA. Period Meter A was bypassed.
7. SUA. Period Meter B was bypassed.

Obviously (13) can't be true if (5) is false, so all probabilities associated with the argument list states that have a zero-state (false) for (5) must be zero—and are therefore not listed. When neither period meter is bypassed, the probability of a scram failure is very low: .001; this, perhaps, could be considered the probability that the circuitry on the output side of the period meters fails. When only one period meter is bypassed, the slightly higher probability, .002, might reflect an additional probability of failure within the meter itself. (The probability of independent failure of two meters was negligibly small compared to the .001 that represents possible failure beyond the period meters.) Finally, if both period meters are bypassed, the probability of a scram failure is 1.0. This implies that a startup accident is defined in such a way that the period scram should always be tripped. The number that appears under "Arg State" is readily identified with the bit configuration under the argument list; e.g., 110 in binary is 6 in hexadecimal.

Zero Entries by Judgment

Consider proposition (24) and note that entry 11100 (Hex 1C) is listed with a probability of 0.5. Almost certainly one would expect the probability of melting to be at least as great or greater for argument states 11101 (1D), 11110 (1E), and 11111 (1F). The omission of these states from the data list means they will be considered as having zero probability. This might be explained as a case where the analyst judged that contributions by these additional terms would have to be small compared to that from the 1C term. To demonstrate the maximum effect on the
results, the problem was rerun, but with the three additional terms included — each with a probability of 1.0. The rerun yielded an absolute probability of $3.3 \times 10^{-6}$ for (25) instead of the original $2.7 \times 10^{-6}$. The difference is indeed negligible, and the omission of the data is justified in this case.

Segmentation

The output below the Summary of Appendix E indicates the sample problem was solved in 5 subset calculations and that the maximum subset size was 16. The segmentation algorithm caused the computation to proceed as follows (refer to the listing of the NCV array):

1. The first actionable subset consisted of the ten bivariates 1, 2, 3, 4, 19, 20, 21, 22, 23, and 24. External bivariates were 5, 12, and 25. Only 25 is isolated and therefore reducible. Bivariates 19, 20, 21, 22, 23, and 24 are resolvable; 1, 2, 3, and 4 are not. Reduction of 25 in this case leaves a null argument list.

2. The second actionable subset consisted of the single bivariates 25 because its argument list is null and can be resolved and because it is now the shortest actionable subset.

3. The third actionable subset consisted of the 16 bivariates 1 through 16. The only external (and therefore isolated) bivariates is 17. In the computation, 1 through 16 were resolved and 17 was reduced to a null argument list.

4. The fourth actionable subset consisted of a single bivariates 17. There was a single external bivariates, 18. On computation, 17 was resolved and 18 was reduced to null argument list.

5. The fifth and last subset consisted of a single bivariates, 18. There were no external bivariates, but the subset was nevertheless considered actionable because it was possible to resolve at least one bivariates (18).

With this segmentation, the execution time on the UNIVAC 1108 and on the IBM System/360-65 was approximately 6 seconds. In an earlier version with less powerful segmentation, the sample problem was computed as a single subset containing bivariates 1 through 24, with 25 as external; execution time was 47 seconds. Although this difference in time is not important for the sample problem, the improved segmentation might make the difference between minutes and hours in the solution of larger problems.
# APPENDIX A

## Card Input Data

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column 6</th>
<th>Column 7</th>
<th>Column 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

- 41 -
APPENDIX B

Sample Problem with 25 Bivariates

Input Data

<table>
<thead>
<tr>
<th>VAR</th>
<th>NUMBER OF CONTINGENTS</th>
<th>DATA ENTRIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

APPENDIX C

NCV Array

1 0
2 1
3 1 2
4 3
5 1 4
6 5
7 5 6
8 5
9 5 8
10 5 8 9
11 5
12 4 5
13 5 6 7
14 5 8 9 10
15 11 12 13 14
16 12 15
17 12 16
18 17
19 1 3 4
20 19
21 19 20
22 19
23 19
24 4 20 21 22 23
25 4 24

- 42 -
APPENDIX D

KSTR Array

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>17</td>
<td>12</td>
<td>18</td>
<td>13</td>
<td>20</td>
<td>14</td>
<td>24</td>
<td>15</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>38</td>
<td>17</td>
<td>40</td>
<td>18</td>
<td>41</td>
<td>19</td>
<td>42</td>
<td>20</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>45</td>
<td>22</td>
<td>47</td>
<td>25</td>
<td>48</td>
<td>24</td>
<td>49</td>
<td>25</td>
<td>55</td>
<td>55</td>
</tr>
</tbody>
</table>

KHEX Array

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>14</td>
<td>15</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>19</td>
</tr>
<tr>
<td>28</td>
<td>24</td>
<td>22</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PROST Array

0.75E 00 0.10E 00 0.50E-01 0.10E 00 0.10E-01 0.10E-01 0.50E 00 0.50E-01 0.50E-01 0.10E 00 0.10E-01 0.10E-01
0.10E 00 0.10E-01 0.10E 00 0.10E-01 0.10E-01 0.10E-02 0.10E-01 0.10E-01 0.10E-02 0.10E 00 0.10E-01 0.10E-02
0.20E-02 0.20E-02 0.10E 01 0.10E-03 0.10E-03 0.10E-03 0.10E 01 0.10E-01 0.10E-01 0.10E-01 0.10E-01 0.10E-01
0.10E 01 0.10E-02 0.10E-01 0.10E 00 0.10E-01 0.10E-01 0.10E 00 0.10E-01 0.10E-01 0.10E-01 0.10E-01 0.10E-01
0.20E 00 0.10E-01 0.10E 00 0.10E 00 0.10E 00 0.50E 00 0.10E 00 0.10E-01 0.10E-01 0.10E 00 0.20E 00 0.10E 01
0.50E 00 0.50E-01 0.10E-01 0.10E-01 0.30E-01 0.50E 00

- 43 -
## APPENDIX E

### Summary of Results

<table>
<thead>
<tr>
<th>BIVAR</th>
<th>PROB</th>
<th>RUE</th>
<th>RISK</th>
<th>CUMRISK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.75E 00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.75E-01</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.41E-01</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.21E-02</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.10E-02</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>0.52E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>0.49E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>0.10E-03</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>0.94E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>0.85E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11</td>
<td>0.10E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>12</td>
<td>0.10E-03</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>13</td>
<td>0.16E-05</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>14</td>
<td>0.13E-06</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>15</td>
<td>0.18E-07</td>
<td>0.10E 01</td>
<td>0.18E-07</td>
<td>0.18E-07</td>
</tr>
<tr>
<td>16</td>
<td>0.84E-08</td>
<td>0.10E 02</td>
<td>0.84E-07</td>
<td>0.10E-06</td>
</tr>
<tr>
<td>17</td>
<td>0.83E-09</td>
<td>0.50E 03</td>
<td>0.42E-06</td>
<td>0.52E-06</td>
</tr>
<tr>
<td>18</td>
<td>0.17E-09</td>
<td>0.50E 03</td>
<td>0.83E-07</td>
<td>0.60E-06</td>
</tr>
<tr>
<td>19</td>
<td>0.60E-03</td>
<td>0.0</td>
<td>0.0</td>
<td>0.60E-06</td>
</tr>
<tr>
<td>20</td>
<td>0.60E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.60E-06</td>
</tr>
<tr>
<td>21</td>
<td>0.84E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.84E-04</td>
</tr>
<tr>
<td>22</td>
<td>0.60E-04</td>
<td>0.0</td>
<td>0.0</td>
<td>0.60E-06</td>
</tr>
<tr>
<td>23</td>
<td>0.60E-05</td>
<td>0.0</td>
<td>0.0</td>
<td>0.60E-06</td>
</tr>
<tr>
<td>24</td>
<td>0.58E-05</td>
<td>0.10E 02</td>
<td>0.58E-04</td>
<td>0.58E-04</td>
</tr>
<tr>
<td>25</td>
<td>0.27E-05</td>
<td>0.10E 03</td>
<td>0.27E-03</td>
<td>0.33E-03</td>
</tr>
</tbody>
</table>

**Subset Calculations:** 5  
**Max Size Subset:** 16
APPENDIX F

FORTRAN Listing

PROBABILISTIC ANALYSIS OF RISKS
ADAPTED FROM PARA WRITTEN BY J. A. CRUACH JUNE 1967
CONTAINS MODIFICATIONS IN LBCS SUGGESTED BY CRUACH
AND MODIFICATIONS IN READ WRITE FORMATS WRITTEN
BY L. M. ARNETT AUGUST 1969
COMMON SAVE/KEY(1000),RDATE(1000),KSTR(2001),GCV(2001),32,
IPREST(3000),KHEX(3000)
COMMON RUE(1000),NUMP(1000),AASPRU(1000),NAPI(1000),NIP(1000),
UPU(2001),NUM(1000),PMLU(2001),GUKR(1000),RISK(1000)
COMMON PKCM(1000),NFAM(1000),NMA(1000)
REAL N30, N10
NIN=5
NUT=6
KOP=0
NLU=1.0E-30
READ (NIN,1)
WRITE (NUT,1)
1 FORMAT (72H1)
1
DU 2 I=1,200
AASPRU(I)=0.
NFAM(I)=0
NUP(I)=0
NIP(I)=0
NUMP(I)=0
KSTRI(I)=0
KSTR(I)=0
DU 2 J=1,32
2 NCV(I,J)=0
DU 3 J=1,5000
PHUST(I)=0.
PKCM(I)=0.
3 KHEXI(I)=0.
READ(NIN,6)NUT,KOP,MAX,KNU
6 FORMAT (15I5)
IF (MAX=49) GO TO 23
MAX=23
MAK=O.
DU 1 J=1,4139
READ(NIN,5)TINS,TINF,KC(I),KE(I),NCV(I,J),J=1,9
5 FORMAT (15I5)
IF (NCV(I,J)=0) GO TO 201
IF (KE(I)=0) GO TO 200
READ(NIN,21)NCV(I,J),J=1,10,KE
200 FORMAT (1B15)
CONTINUE
1 I=1 WHERE NMA(I) GO TO 15
KSTRI(I)=KE
KSTR(I)=MAK+1
IF (INFOMKE+1) GO TO 8
LEN=3*KK
KAL=MAK+1
NAD=MAK+1
LEN=LEN+1
PK(KE(I),J)=KK
8 FORMAT (15E5.2)
I4=0.
DU 4 K=1,LEN
NAD=MAK+1
IF (PK(KAM,K)=(LEN+1)) GO TO 4
I4=I4+1
KAD=KAD+1
KHEX(KEI)=K-1

- 45 -
KDATA(1) = 1A
K6 = MARK*K
PRUST(KK3) = PRUST(KK4)
4 CONTINUE
LEN $ = 1A
OU TU 10
8 KDATA(1) = KA
KA1 = MARK+1
KA2 = MARK+KA
IF (KDATA KA1 $) OU TU 225
READ (NIN=5) KMEK(J,J),PRUST(J),J = KA1,KA2
5 FORMAT (5I10,E5.2,5X)
OU TU 221
225 READ(NIN,222)KMEK(J,J),PRUST(J),J = KA1,KA2
222 FORMAT(5(10,E5.2,5X)
221 LEN = KA
10 MARK = MARK * LEN
OU TU 20
15 WRITE(NOUT,16)
16 FORMAT(///,30X,17HINPUT ERROR, STOP)
STOP

C
20 WRITE(NOUT,17)
17 FORMAT (95X,10HINPUT DATA/55X,10H--------)
WRITE(NOUT,18)
18 FORMAT (//20X,46MVAK NUMBKR OF CONTIGUOUS DATA ENTRIES/
122X4OM== -----------------------------------------------)
OU 19 I = 1,NTUT
19 WRITE(NOUT,14) 1,HREV1),KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
14 FORMAT(25X,10X,KJAT(I)
601 CONTINUE
   GO TO 25
603 JUP = J
   CALL FAM(JUP,NIP)
   DU 604 I=1,NUT
   IF(NIP(I) .EQ. 1) NUP(I) = 1
604 CONTINUE
   GO TO 600
C NFAM=1 MARKS FAMILY GENERATORS
25 NUO = 10000
   IT=NUT
   LAST=0
   KAN=0
C SELECT TRIAL SUBSET WITH MAX INDEX KAN
27 KAN=KAN+1
   IF(KAN .LE. NUT) GO TO 30
   IF (NUO .EQ. 10000) GO TO 100
   GO TO 55
30 IF(NUM(KAN) .EQ. 1) GO TO 27
   DU 32 I=1,NUT
32 KIP(I) = 0
   CALL FAM(KAN,NET)
C CALCULABLE SUBSET IS NOW INDICATED BY NET EQ 1.
C SUBSET BASED ON KAN. TEST TO SEE IF ACTIONABLE
33 NCT =0
   NAK=1
   DU 40 I=1,NUT
   IF (NET(I) .EQ. 1) NCT =NCT+1
   IF (NET(I) .EQ. 1) GO TO 40
   IF (NUM(I) .EQ. 1) GO TO 40
   NUM=KEV(I)
   IF (NUM .EQ. 0) GO TO 40
   DU 33 J=1,NUM
   KA = NCV(I,J)
   IF(NET(KA) .EQ. 1) NET(I) =2
   IF(KAN .EQ. KA) NAK=0
33 CONTINUE
40 CONTINUE
C EXTERNAL MEMBERS ARE NET=2
   IF (NCT .GE. NUO .AND. LAST .GE. 0) GO TO 27
   DU 48 I=1,NUT
   IF (NFAM(I) .NE. 1) GO TO 48
   CALL FAM(I,NUP)
   KILL =0
   DU 41 J=1,NUT
   IF(NET(I) .NE. 2) GO TO 41
   IF(NUP(I) .NE. 1) GO TO 41
   KILL =KILL+1
   IF(KILL .LE. 1) DU TO 42
   NIP(IKUOU)=2
   NIP(I) = 2
42 JRUU = J
41 CONTINUE
48 CONTINUE
   NUK = 0
   DU 50 I =1,NUT
   IF (NET(I) .NE. 2) GO TO 50
   IF (NIP(I) .EQ. 2) GO TO 50
   NIP(I) = 4
   NUK=1
50 CONTINUE
   IF (NUK .EQ. 1) GO TO 54

- 47 -
IF (NAX .LE. 1) GO TO 54
SUBSET NOT ACTIONABLE. LOOP AGAIN
GO TO 27
C
CANDIDATE, COUNT MEMBERS AS NCT
54 IF (NCT .GE. 11) GO TO 60
IF (LAST .EQ. 11) GO TO 60
C
SAVE NCT AS NLU IF IT IS LOWEST YET FOUND
IF (NCT .GT. NLU) GO TO 27
NLU = NCT
IF = KAN
GO TO 27
55 LAST = 1
KAN = IF
GO TO 30
C
SUBSET OF LOWEST INDEX KAN WITH NCT
C
MEMBERS NOT TO BE COMPUTED. MEMBERS ARE GIVEN AS NCT EQUAL 1.
C
INSTALL IN NGRP
60 NGRP = 0
GO 62 IF = 1, KAN
IF (NIP(I) .NE. 1) GO TO 62
NGRP = NGRP + 1
NGRP = NGRP
62 CONTINUE
C
IDENTIFY IN NIP THE FOLLOWING CODES
C
NIP = 0 IGNURE
C
NIP = 1 UNRESOLVABLE
C
NIP = 2 IRREDUCIBLE
C
NIP = 3 RESOLVABLE
C
NIP = 4 REDUCIBLE
C
NIP=0
GO 65 I = 1, NIP
IF (NIP(I) .EQ. 4) GO TO 65
IF (NIP(I) .EQ. 1) GO TO 65
IF (NIP(I) .EQ. 1) GO TO 65
NUM = KEY(I)
NIP(I) = 0
IF (NUM .EQ. 0) GO TO 65
GO 66 J = 1, NUM
NAC = NCV(I, J)
IF (NAC .NE. 1) GO TO 66
IF (NAC .GT. NUM) NAC = NCV(I, J)
66 CONTINUE
65 CONTINUE
GO 67 J = 1, NCT
KAN = NGRP(J)
IF (KAN .EQ. NUM) GO TO 68
NIP(KAN) = 3
GO TO 67
68 NIP(KAN) = 1
67 CONTINUE
C
CALCULATE SUBSET PROBABILITIES
C
IF (NCT .LT. NMAX) GO TO 810
WRITE (INJUL, (612)) NCT, NMAX
3 FORMAT (13, 1X, 10H0SUBSET SIZE OF, 13, 1X, 28HMAXIMUM MAXIMUM SPECIFIED
1 AS, 13, 36X, 30HTERMINATE WITH PARTIAL SUMMARY,///
GO TO 100!
610 NCALC = NCALC + 1
IF (NCT .LT. MAXSUB1) NMAX = NCT
NUM = KUP
GO TO 347 I = 1, KUP
M = NGRP(I)
NET(M) = 0
- 48 -
347  NB(I) = 0
     K=1
 349  GU TO 34d
 352  IF (NB(IA) .EQ. 0) GU TO 367
 355  NB(LA) = 0
     M=NUMP(LA)
     NET(M)=1
 356  LA=LA+1
     IF(LA.LT.1) GU TO 375
     GU TO 365

 367  NB(LA) = 1
     M=NUMP(LA)
     NET(M)=1
     NUM = KEY(M)
     JX = AXM(M)
     IF(NUM.EQ.0) GU TO 370
     L = 0
     DJ 368 J = J + NUM
     GU = NCVM(J)
     IF (NET(M) .EQ. 1) L = L + 2**((NUM - J)

 368  CONTINUE
     JY = JX + KDAT(M) - 1
     GU 369 J = JA + JY
     IF(L.EQ.KHEX(J)) GU TO 370

 369  CONTINUE
     GU TO 360
 370  PMUL(LA) = PHST(J)
     IF(L.EQ.KHEX(KUP)) GU TO 361
     K = LA + 1
 346  GU 360 1=K,KUP
     PMUL(I) = 1.0
     M = NUMP(I)
     NUM = KEV(M)
     JX = AXM(M)
     J = JX
     IF(NUM.EQ.0) GU TO 355
     L = 0
     DJ 352 J = J + NUM
     GU = NCVM(J)
     IF (NET(M) .EQ. 1) L = L + 2**((NUM - J)

 350  CONTINUE
     JY = JX + KDAT(M) - 1.
     GU 352 = JX + JY
     IF(L.EQ.KHEX(J)) GU TO 355

 352  CONTINUE
     GU TO 360
 355  PMUL(I) = 1.0 - PHST(J)

 360  CONTINUE
 361  PHU = 1.0
     GU 363 I=1,NUP
 363  PHU = PRU*PMUL(I)

C ASSIGN PROBABILITY INCREMENTS

 365  GU 365 I=1,NUT
     IF(NI(I).LE.1) GU TO 385
     IF(NI(I).NE.3) GU TO 371
     IF(NET(I).EQ.0) GU TO 365
     ABSPRU(I) = ABSPKU(I)*PHU
     GU TO 385
 371  KA = KSAT(I)
     KY = KA + KDAT(I) - 1

- 49 -
NUM = KEV(1)
LOOK = 0
JB = 0
DU 376 J=1,NUM
LU = NCV(1,J)
IF(LU=JU) WW TU 380
IF(NIP(1)),EVE OU DU TU 376
JUT = 2** (NUM-J)
JB=JB+JUT
IF (NET(1),EVE ) OU TU 376
LOOK=LOOK + JUT
376 CONTINUE
380 DU 382 L=XX,XY
JA = KHEX(L)
JC=LOLAND(JA+JU)
IF(JC<>NE,LOOK OU TU 382
PKML(I) = PKCM(L) + PRJ0
382 CONTINUE
385 CONTINUE
WW TU 349
C CONSOLIDATE KHEX AND PKUST ARRAYS
C RESET KEV, KDAT AND NCV
375 DU 395 J = 1, I, JUT
IF(INP(IJ) ,EVE ) NUM(I,J) = 1
IF(INP(IJ) ,NE 4) OU TU 395
JX = AXRI(IJ)
JY = JX + KJAT(I) - 1
NUM = KEV(I)
DU 391 J = 1, NUM
LU = NCV(I,J)
IF(NIP(LU) ,NE 0) OU NCV(I,J) = 0
391 CONTINUE
DU 405 L=JA,JY
PKUST(L) = PKUST(I) + PKCM(L)
PKML(I) =jure
JA = KHEX(L)
JD = 1
JJ = 1
JL = O
DU 410 J=1, NUM
LA = NUM-JT]
IF (NCV(I,JL),EVE ) OU TJ 410
IF (LOLAND(JA,JL),EVE ) OU TJ 409
JL=JL+JC
409 JD=2*JD
410 JL=2*JL
405 KHEK(I) = JJ
LA=NUM
L=O
DU 411 J=1, NUM
IF(NCV(I,J),NE 0) OU TU 415
L=L+1
NCV(I,L) = NCV(I,J)
DU TU 411
415 LA=LA-1
411 CONTINUE
KEV(I)=LA
DU 388 J = JX,JY
IF (PKUST(J) ,LT. N2U) OU TJ 388
LA = KHEX(J)
IF (L <>EVE , JY) OU TU 388
JN = J + 1
100 SUM = 0
101    MSTR(I1) = SUM
102      ABSK1I1 = PRST(I1)
103      SUM = SUM + MSTR(I1)
104     C101N(I1) = SUM
105 WRITE RESULTS
106 FORMAT (15X,'SUMMARY/55X,7H=***1')
107 WRITE(NOUT,107)
108 FORMAT (15X,'MSTR(10) PHUO  KUE  K13  CUM13K1/1***1')
109 WRITE(NOUT,109)
110 WRITE(NOUT,110)
111 END
SUBROUTINE FAM FOR PAR4
SUBROUTINE FAM(KAN, NET)
COMMON/SAVE/KEV(200), KDAT(200), KSTR(200), NCY(200, 32),
IPRST(3000), KHEX(3000)
INTEGER NET(200)
DU 32 I = 1, 200
32 NET(I) = 0
NET(KAN) = 1
NUM = KEV(KAN)
IF (NUM.EQ.0) GO TO 40
DU 35 J = 1, NUM
KA = NCY(KAN, J)
35 NET(KA) = 1
KANDN = KAN - 1
DU 38 I = 1, KANDN
IA = KAN - I
IF (NET(IA) .EQ. 0) GO TO 38
NUM = KEV(IA)
IF (NUM .EQ. 0) GO TO 38
DU 39 J = 1, NUM
KA = NCY(IA, J)
39 NET(KA) = 1
38 CONTINUE
40 RETURN
END