DP-1207

AEC RESEARCH AND DEVELOPMENT REPORT

QUANTITATIVE SAFETY ANALYSIS

J.W. CROACH AND L.M. ARNETT

DU PONT
REG. U. S. PAT. OFF

Savannah River Laboratory

Aiken, South Carolina

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

663764

DP-1207

Mathematics and Computers
(TID-4500, UC-32)

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

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 bivariables. The bivariables 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 bivariables 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 $Z_{5,0}$, $Z_{5,1}$, $Z_{5,2}$, ..., $Z_{5,31}$.

The set of bivariables (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 bivariables 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 bivariable 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, ..., 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

$$P\ \{A \text{ x } B\} = P\ \{B|A\}\ P\ \{A\} \tag{3}$$

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

$$P\ \{H_2 \text{ x } H_1\} = P\ \{H_2|H_1\}\ P\ \{H_1\}$$

$P\ \{H_1\}$ 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:

$$P\ \{A \times B \times C\} = P\ \{C|A \times B\}\ P\ \{B|A\}\ P\ \{A\} \tag{4}$$

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

The ordered sequence of S and Z sets is

S_1, S_2, S_3, ... S_n and Z_1, Z_2, Z_3, ..., Z_n

Each set in the two sequences is generated by adding another bivariable, say X_m, to the m-1 bivariables 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:

$$P\ \{(X_m = 1) \times Z_{m-1,j}\} = P\ \{(X_m = 1)|Z_{m-1,j}\}\ P\ \{Z_{m-1,j}\} \tag{5}$$

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

$$Q_{mi} = P_{mj} \cdot Q_{m-1,j} \tag{6}$$

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:

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

$$Q_{m,2i} = (1 - P_{mi})Q_{m-1,i} \qquad (7)$$

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_n 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 bivariables (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 bivariables 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 bivariable interdependence
2. Exploitation of a high proportion of zero entries in the input data
3. Segmentation of a large bivariable set into loosely coupled subsets

Each of these concepts is discussed in the following sections.

Implicit Bivariable Interdependence

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

$$P_{mi} = P\ \{(X_m = 1) | Z_{m-1,i}\} = \frac{P\{(X_m = 1)\ x\ Z_{m-1,i}\}}{P\ \{Z_{m-1,k}\}} = F_m(Z_{m-1,i}) \tag{5}$$

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

$$P_{mkj} = F_{mk}(Z_{kj}) \tag{8}$$

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 bivariable 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, and X_9 of the set (m-1 = 10), X_1, X_2, ..., X_{10} will generate a sequence of binary numbers $Z_{31} = 0$, $Z_{32} = 1$, ..., $Z_{38} = 111$. 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_{mo}.

The functional dependence given in equation (8) implies that the k bivariables 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 bivariable 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 $2^{10} = 1024$ values in the full function set for X_{11} implied by equation (5).

Zero Entries

Even though all possible bivariables 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 bivariable X_m may include ten earlier bivariables 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 bivariables in the argument list have the value 1 (yes). 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. Bivariables 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 bivariable permits retrieval of the matching probability; failure to find an identifier in its appropriate array means that the corresponding probability is zero. Computational advantages are more apparent in the subsequent discussion of the computer program.

Segmentation

The computing time for a problem with n bivariables may be roughly proportional to the number of possible states that must be considered; i.e., Z_n, or approximately 2^n. If such a problem could somehow be reduced to the solution of two independent problems, each comprising n/2 bivariables, 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^{16}! 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.

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 bivariables has been defined, and the argument list for each bivariable is specified. Any bivariable 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 bivariables 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'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 bivariables 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 bivariable in a chain, and there may be as many as n bivariables.

It is now possible to deal more precisely with interaction. Two bivariables 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 bivariables 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 bivariable 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 bivariable in the subset contains within its argument list (if any) only bivariables that are also members of the same calculable subset. Since the indices of bivariables in any argument list are required to be less than the index of the bivariable itself, at least one bivariable of a calculable subset must have a null argument list. In other words, the absolute, unconditional probability must be given for at least one bivariable in a calculable subset.

Examples of calculable subsets are (each bivariable is represented by its index):

Example 1

Bivariable	Argument List
1	Null
2	1
3	1,2
4	1,3
10	2,3,4

Example 2

Bivariable	Argument List
6	Null
7	6
8	7
9	8

The following subset is not calculable:

Bivariable	Argument List
12	10,11
13	12
14	13
15	12,13,14

In the last example, the subset is not calculable because of the presence of bivariables 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_n has been somehow selected and that the interaction of A with the remainder of S_n is to be considered. (The intent is to determine whether the subset A may be usefully segmented from S_n for computational purposes.) Define another subset E in S_n that will be called the set of external bivariables associated with A. An external bivariable (by definition): (1) is not contained in A and (2) contains within its argument list at least one bivariable that belongs to A.

If the set of external bivariables, 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 bivariables 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 $\tilde{A}$.) A measure of the degree of interaction between A and $\tilde{A}$ can be gained from the intersection of all chains based in E, i.e., the intersection of all chains based on the external bivariables 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 bivariables within A have an intersection anywhere in $\tilde{A}$ except in E. If B is null, none of the bivariables in E interact with each other. The interaction of A with the individual bivariables 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 bivariables in A are determined.

2. The remaining subset $\tilde{A}$ is free of any residual interactions among bivariables of A.

If B is not null, e.g., some bivariables of E interact, computation of the calculable subset will not permit resolution of all interactions between A and $\tilde{A}$. The bivariables 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 bivariable (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 bivariables 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 bivariables of E at the time the computation of A is carried out. Suppose X_k is a particular isolated bivariable of E. Let the function set for X_k be given as

$$P_i = F_k(Y_A, Y_{\tilde{A}}) \tag{9}$$

where Y_A and $Y_{\tilde{A}}$ denote the two subsets of X_k argument list that contain respectively the bivariables in A and in $\tilde{A}$. 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,

$$P_i = F'_k(Y_{\tilde{A}}) \tag{10}$$

If Y_A and $Y_{\tilde{A}}$ contain p and q bivariables 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

$$P_j = \sum_{i=o}^{2^p-1} Q_i F_k(Y_{A_i}, Y_{\tilde{A}_j}) = F'_k(Y_{\tilde{A}_j}) \tag{11}$$

The notation here is a bit mixed; Y_{A_i} and $Y_{\tilde{A}_j}$ denote specific state combinations of bivariables in Y_A and $Y_{\tilde{A}}$ 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 bivariables 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 bivariables 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 bivariables has been defined. The 2^n possible states of these bivariables 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 bivariable 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} bivariable will have a yes ($X_k = 1$) for the campaign.

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

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, $\overline{R}$, is defined as

$$\overline{R} = \sum_{j=o}^{2^n-1} C_j Q_{nj} \tag{12}$$

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 considered 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

$$C_j = \sum_{i=1}^{n} X_{ij} R_i \tag{13}$$

In equation (13) as before, X_{ij} is assigned the value (0 or 1) that X_i has in the Z_{nj}th state. Combining equation (12) and (13) gives

$$\overline{R} = \sum_{j=o}^{2^n-1} X_i Q_{nj}$$

but

$$\sum_{j=o}^{2^n-1} X_i Q_{nj} = W_i$$

therefore

$$\overline{R} = \sum_{i=1}^{n} W_i R_i \tag{14}$$

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 bivariables. 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 bivariable 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.

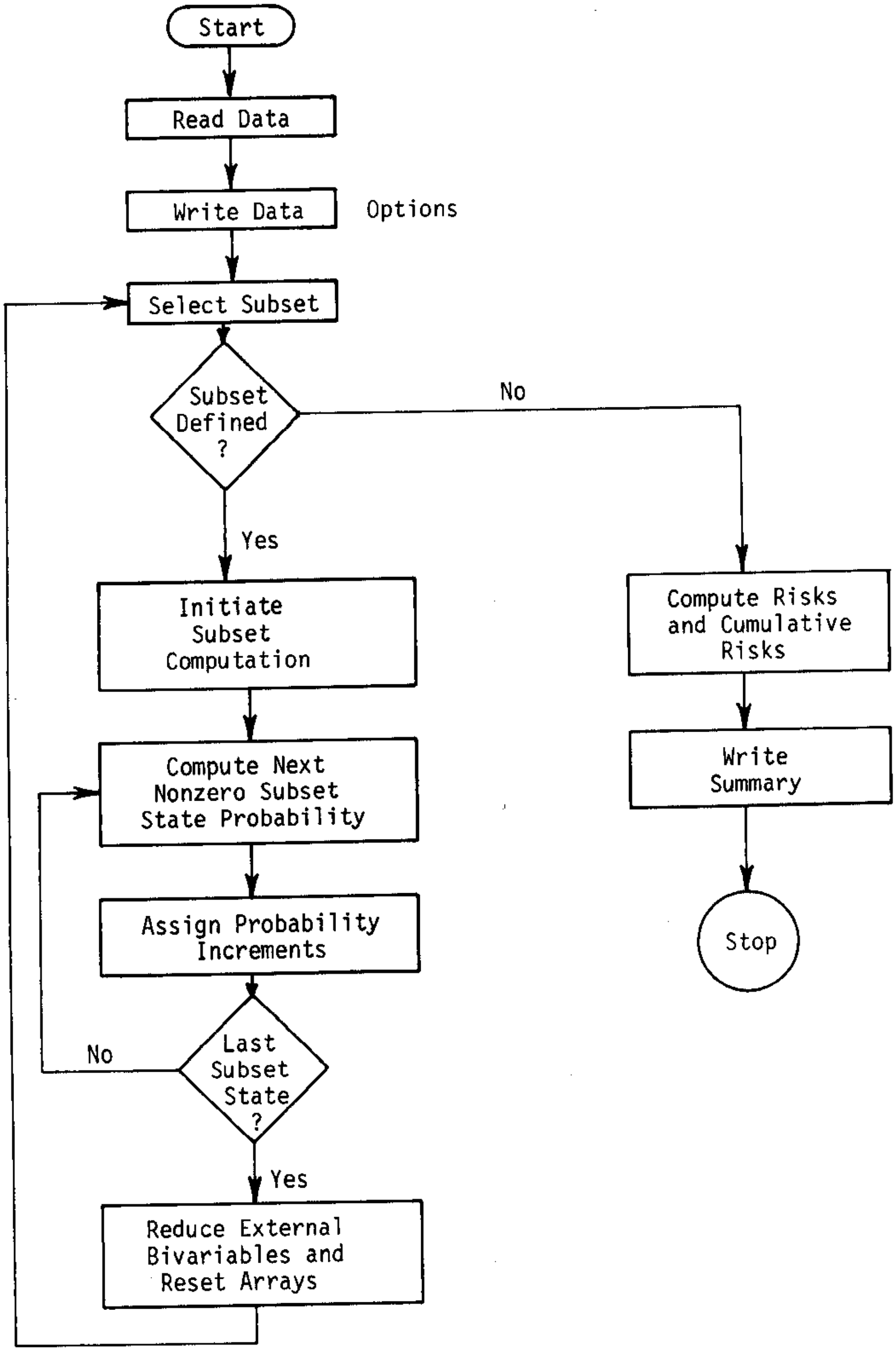
PAR COMPUTER PROGRAM

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 bivariables. We assume that the analyst groups his bivariable 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 Next Nonzero Subset State Probability
Assign Probability Increments
Last Subset State ?
No
Yes
Reduce External Bivariables and Reset Arrays
Compute Risks and Cumulative Risks
Write Summary
Stop

Data Specification

The principal working arrays and mnemonics are:

Mnemonic	Description
KEV(200)	KEV(I) is the number of bivariables in the argument list of the I^{th} bivariable.
NCV(200,32)	NCV(I,J) identifies the J^{th} bivariable in the argument list of I. J < I.
KSTR(200)	KSTR(I) is the address of the initial entry for function set data in KHEX and PRØST for the I^{th} bivariable.
KDAT(200)	KDAT(I) is the number of data pairs entered in KHEX and PRØST for the I^{th} bivariable, beginning at KSTR(I).
KHEX(3000)	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).
PRØST(3000)	PRØST(K) is the probability that the pertinent bivariable is 1 (yes) if the state of its argument list is given by KHEX(K).
RUE(200)	RUE(I) is the relative undesirability estimate for the occurrence of the one-state of the I^{th} bivariable.

NET(200)
NIP(200)
NUP(200)
	Scratch-pad arrays for developing structural relationships.

NGRP(200)	NGRP(J) is the bivariable index of the J^{th} member of the current actionable subset.
NØW(200)	NØW(M) = 0 or 1 indicates bivariable status as respectively unresolved or resolved.
NB(200)	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.

Mnemonic	Description
PMUL(200)	In computing the nonzero probability PRØB of a specific subset state, PMUL(I) is the probability that the Ith bivariable has a one-state.
PRCM(3000)	PRCM(J) accumulates the probability factor that is applied to PRØST(J). This takes place in the reduction of external variables. See Logic.
ABSPRØ(200)	ABSPRØ(I) is the absolute or unconditional probability that I has a one-state (yes) for the campaign.
RISK(200)	At output, RISK(I) = ABSPRØ(I) * RUE(I)
CURISK(200)	At output, CURISK(I) is the accumulated sum of RISK(J) for J < I.
NFAM(200)	NFAM(I) = 1 denotes a family generator. See Logic.
NCT	Number of bivariables in the current actionable subset.
NZØ	NZØ = 10^{-30}; in testing for zero in entries of the PRØST array, any number less than 10^{-30} is considered as zero.
NVAR	Bivariable index on input card.
NTØT	Total number of bivariables defined at input.
NFØRM	Specifies form of input data for a particular bivariable. See Input.
KØPT	Options for listing contents of working arrays. See Input.
NMAX	Maximum permissible subset size. Unless intentionally overridden on input, NMAX is 25.

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 (binary 11001)
J = 21 (binary 10101)
LGLAND(I,J) = 17 (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(I) = 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 Bivariables.

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

It is possible to identify the isolated (if any) external bivariables 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 bivariable. A calculable subset is considered to be actionable if at least one external bivariable is isolated. (It is also possible for a subset to contain a single resolvable member with no external bivariables 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 bivariables 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:

Z	NB(NCT)	NB(NCT-1)	NB(NCT-2)	...
0	0	0	0	
1	1	0	0	
10	0	1	0	
11	1	1	0	
100	0	0	1	
⋮				

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 k^{th} bit. Then there are two rules that govern the selection of Z_{i+1}, depending upon whether the bivariable corresponding to the k^{th} 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 bivariable represented by the k^{th} bit is specified by bits of Z_i to the left of the k^{th} 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 k^{th} and leftward positions. The next potentially nonzero Z in the sequence is therefore obtained from Z_i by arithmetically inserting 1 in the k^{th} 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 bivariables is m instead of 2^m.

When a nonzero state probability for A is found and computed as PRØB, the increment is accumulated in the array ABSPRØ(I) for all resolvable bivariables in subset A; i.e., the current value of PRØB is added to ABSPRØ(I) if I represents a resolvable bivariable with a one-state in the current Z state for A. The accumulated value of ABSPRØ(I), when the subset computation is complete, is the unconditional probability that the I^{th} bivariable has a value of 1 for the campaign.

Reduction of External Bivariables

The isolated external bivariables in E are prepared for reduction as the subset computation for A proceeds. The current value of PRØB is appropriately accumulated in PRCM(J), where J matches the position of an entry in the array PRØST(J) that stores the input probability function sets for all bivariables. 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 bivariables in its bivariable list; let m be the number of these bivariables in A, and k be the number in $\tilde{A}$. Make the following definitions with respect to the bivariables of the argument list:

Z_T = n-bit binary state identifier for the argument list

Z_A = m-bit binary state identifier for the bivariables in A

$Z_{\tilde{A}}$ = k-bit binary state identifier for bivariables in $\tilde{A}$

Z_A is the binary number obtained by suppressing or ignoring the m bits in Z_T related to $\tilde{A}$; $Z_{\tilde{A}}$ is the binary number similarly derived from Z_T by ignoring the bits related to A. Thus for any specific Z_T there corresponds a unique pair of Z_A and $Z_{\tilde{A}}$. 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 Z_T = 11010,

then

$$Z_A = 1\cancel{1}01\cancel{0} = 101$$

and

$$Z_{\tilde{A}} = \cancel{1}1\cancel{0}\cancel{1}0 = 10$$

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

PRØST(J) = PRCM(J) * PRØST(J)

The binary identifier, Z_T, previously stored in KHEX(J) is replaced by $Z_{\tilde{A}}$. At this point, there may be several entries in the function set that have the same KHEX values ($Z_{\tilde{A}}$). The function set is scanned to find such multiple entries and to consolidate them in a single entry; the corresponding values of PRØST 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 bivariables in the argument list of a reducible external bivariable J are included in the subset A, the reduction process eliminates all bivariables from the argument list of J, and the single remaining entry in the PRØST array for J represents the absolute probability of that bivariable. The corresponding KEV(J) will become zero (corresponds to a null argument list).

Input

Card input data are arranged as follows:

Card Type	Card Columns	Mnemonic	Format	Description
1	1-72	-	72H	Problem title.
2	1-5	NTØT	I5	Total number of bivariables.
	6-10	KØPT	I5	Option specifying listing of input data.
	11-15	NMAX	I5	Maximum permissible size of subset (If blank, program sets NMAX = 25).
	16-20	KHD	I5	Specifies I/O Format to KHEX (see Card Type 5).
3	1-5	NVAR	I5	Index number of bivariable (must follow serial order I = 1, 2, 3, ... NTØT).
	6-10	NFØRM	I5	Specifies card type that follows current Card 3 or its continuation. Type 4 for NFØRM = 0; Type 5 for NFØRM = 1.
	11-25	RUE(I)	E15.5	Relative undesirability of Ith bivariable.
	26-30	KE	I5	Number of bivariables in argument list of I.
	31-35	KA	I5	Number of data pairs for I (needed only for NFØRM = 1).
	36-80	NCV(I,J)	11I5	Bivariable indices in argument list of I, monotonically increasing order.
3'	1-80	NCV(I,J)	16I5	Continuation of 3 if needed.
3''	1-25	NCV(I,J)	5I5	Continuation of 3' if needed.
4	1-80	PRØST	16E5.2	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 NFØRM = 0 on 3.

Card Type	Card Columns	Mnemonic	Format	Description
5	1-10	KHEX	I10 Z10	Type 5 must follow 3 if NFØRM = 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).
	11-15 etc. thru 75	PRØST	E5.2	

Repeat Types 3/4 or 3/5 for each I until all NTØT bivariables are listed.

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.

Input Limitations

Maximum number of bivariables (Working arrays could be redimensioned to accommodate many more bivariables if desired.)	200
Maximum number of bivariables in any argument list	32
Maximum number of total pairs of entries in KHEX and PRØST	3000

Options

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

KØPT	Output List
0	For each bivariable index I, KDAT(I) and KEV(I) are listed
1	As above plus listing of NCV array
2	Same as 0 plus listing of KSTR and KHEX
3	Combine 0, 1, and 2
6	Combine 0, 2 plus PRØST array
7	Combine 0, 1, 2 plus PRØST

SAMPLE PROBLEM

A short sample problem is described in detail to illustrate the application of PAR. The problem comprises a set of 25 bivariables 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 bivariables 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.

				Data	
No.	Bivariable Proposition			Arg State	Prob
1	The reactor operated.				
	Arg List:	Null			
	RUE = 0			-	.75
2	Departure from standard operation was required.				
	Arg List:	1			
	RUE = 0	1		1	.10
3	There was an operating error.				
	Arg List:	1	2		
	RUE = 0	1	0	2	.05
		1	1	3	.10
4	There was a serious operating error.				
	Arg List:	3			
	RUE = 0	1		1	.05
5	A startup accident occurred (SUA).				
	Arg List:	1	4		
	RUE = 0	1	1	3	.50
6	SUA. Period meter "A" was bypassed.				
	Arg List:	5			
	RUE = 0	1		1	.05
7	SUA. Period meter "B" was bypassed.				
	Arg List:	5	6		
	RUE = 0	1	0	2	.05
		1	1	3	.01
8	SUA. Flux monitor "C" was bypassed.				
	Arg List:	5			
	RUE = 0	1		1	.10
9	SUA. Flux monitor "D" was bypassed.				
	Arg List:	5	8		
	RUE = 0	1	0	2	.10
		1	1	3	.01

No.	Bivariable Proposition		Data	
			Arg State	Prob
10	SUA. Flux monitor "E" was bypassed.			
	Arg List:	5 8 9		
	RUE = 0	1 0 0	4	.10
		1 0 1	5	.01
		1 1 0	6	.01
		1 1 1	7	.001
11	SUA. Flux trip level set too high.			
	Arg List:	5		
	RUE = 0	1	1	.10
12	SUA. Reactivity addition rate exceeded x % k per second.			
	Arg List:	4 5		
	RUE = 0	0 1	1	.01
		1 1	3	.10
13	SUA. Period scram failed			
	Arg List:	5 6 7		
	RUE = 0	1 0 0	4	.001
		1 0 1	5	.002
		1 1 0	6	.002
		1 1 1	7	1.0
14	SUA. Flux level scram failed.			
	Arg List:	5 8 9 10		
	RUE = 0	1 0 0 0	8	.0001
		1 0 0 1	9	.0001
		1 0 1 0	A	.0001
		1 0 1 1	B	.01
		1 1 0 0	C	.0001
		1 1 0 1	D	.01
		1 1 1 0	E	.01
		1 1 1 1	F	1.0
15	SUA. Some fuel was melted.			
	Arg List:	11 12 13 14		
	RUE = 1	0 0 1 0	2	.001
		0 0 1 1	3	.01
		0 1 1 0	6	.10
		0 1 1 1	7	.50
		1 1 1 0	E	.20
		1 1 1 1	F	1.0
16	SUA. More than 10% of fuel melted.			
	Arg List:	12 15		
	RUE = 10	0 1	1	.05
		1 1	3	.50

		Data	
No.	Bivariable Proposition	Arg State	Prob
17	SUA. More than 50% of fuel melted.		
	Arg List: 12 16		
	RUE = 500 1 1	3	.10
18	SUA. Reactor was disrupted.		
	Arg List: 17		
	RUE = 500 1	1	.20
19	A power surge accident occurred (PSA).		
	Arg List: 1 3 4		
	RUE = 0 1 1 0	6	.01
	1 1 1	7	.10
20	PSA. Temperature monitor trip level set too high.		
	Arg List: 19		
	RUE = 0 1	1	.10
21	PSA. Temperature operating limit in use was incorrectly high.		
	Arg List: 19 20		
	RUE = 0 1 0	2	.10
	1 1	3	.50
22	PSA. Unwanted control rod motion occurred.		
	Arg List: 19		
	RUE = 0 1	1	.10
23	PSA. Temperature monitor scram failed.		
	Arg List: 19		
	RUE = 0 1	1	.01
24	PSA. Some fuel melted.		
	Arg List: 4 20 21 22 23		
	RUE = 10 0 0 0 1 1	3	.20
	1 0 0 1 1	13	1.0
	1 1 1 0 0	1C	.50
	1 1 0 0 0	18	.05
	1 0 1 1 0	16	.10
	0 0 0 1 0	2	.01
25	PSA. More than 10% of fuel melted.		
	Arg List: 4 24		
	RUE = 100 0 1	1	.05
		3	.50

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 KØPT = 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 2.7×10^{-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 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 $3.3 \cdot 10^{-6}$ for (25) instead of the original $2.7 \cdot 10^{-6}$. 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 bivariables 1, 2, 3, 4, 19, 20, 21, 22, 23, and 24. External bivariables were 5, 12, and 25. Only 25 is isolated and therefore reducible. Bivariables 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 bivariable 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 bivariables 1 through 16. The only external (and therefore isolated) bivariable 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 bivariable, 17. There was a single external bivariable, 18. On computation, 17 was resolved and 18 was reduced (to null argument list).

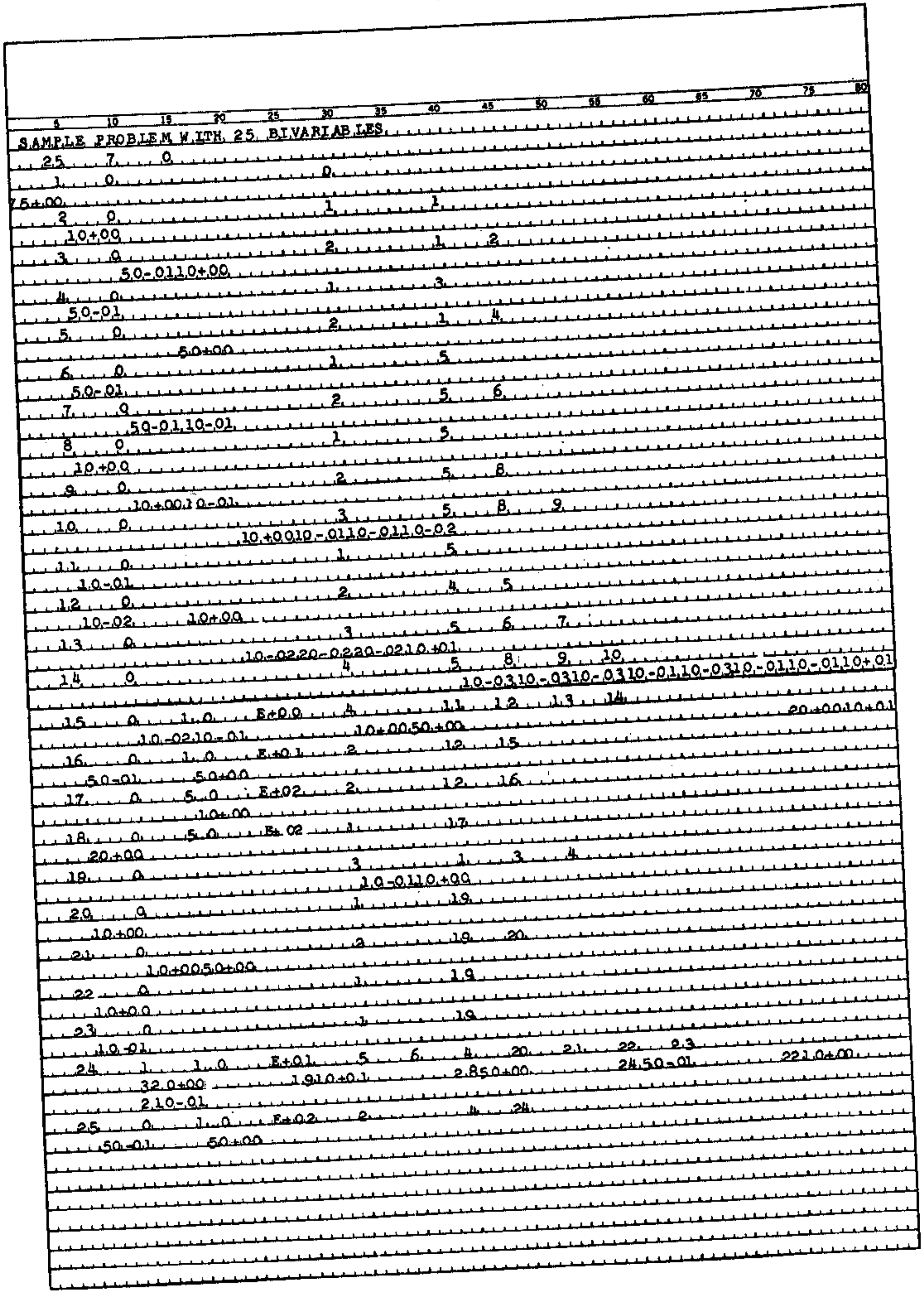
5. The fifth and last subset consisted of a single bivariable, 18. There were no external bivariables, but the subset was nevertheless considered actionable because it was possible to resolve at least one bivariable (18).

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 bivariables 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.

EQN:bch

APPENDIX A

Card Input Data



```
SAMPLE PROBLEM WITH 25 BIVARIABLES.
   25   7.   0.
   1.   0.                  0.
75+00.
   2.   0.                  1.        1.
     10+00.
   3.   0.                  2.        1.   2.
          5.0-0110+00.
   4.   0.                  1.        3.
     5.0-01.
   5.   0.                  2.        1.   4.
               5.0+00
   6.   0.                  1.        5.
     5.0-01.
   7.   0.                  2.        5.   6.
          5.0-01.10-01.
   8.   0.                  1.        5.
     10+0.0
   9.   0.                  2.        5.   8.
          10+0010-01.
  10.   0.                  3.        5.   8.   9.
                    10+0010-0110-0110-02
  11.   0.                  1.        5.
     10-01.
  12.   0.                  2.        4.   5.
     10-02.    10+00
  13.   0.                  3.        5.   6.   7.
                    10-0220-0220-0210+01.
  14.   0.                  4.        5.   8.   9.  10.
                                        10-0310-0310-0310-0110-0310-0110-0110+01
  15.   0.  1.0     E+0.0   4.       11.  12.  13.  14.               20+0010+01
          10-0210-01          10+0050+00
  16.   0.  1.0     E+01    2.       12.  15.
     5.0-01    5.0+00
  17.   0.  5.0     E+02    2.       12.  16.
               1.0+00
  18.   0.  5.0     E+02    1.       17.
     2.0+00
  19.   0.                  3.        1.   3.   4.
                              1.0-0110+00
  20.   9.                  1.       19.
     10+00.
  21.   0.                  2.       19.  20.
          1.0+0050+00
  22.   0.                  1.       19
     1.0+00
  23.   0.                  1.       19.
     1.0-01.                          4.  20.  21.  22.  2.3
  24.   1.  1.0     E+01    5.    6.                  24.50-01       221.0+00
         32.0+00       1.910+01       2.850+00
         2.10-01.
  25.   0.  1.0     E+02    2.        4.  24.
     50-01     50+00
```

APPENDIX B

Sample Problem with 25 Bivariables

Input Data

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

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			

APPENDIX D

KSTR Array

```
    1       1       2       2       3       3       4       5       5       6
    6       7       7       8       8      10       9      11      10      13
   11      17      12      18      13      20      14      24      15      32
   16      38      17      40      18      41      19      42      20      44
   21      45      22      47      23      48      24      49      25      55
```

KHEX Array

```
    0       1       2       3       1       3       1       2       3       1
    2       3       4       5       6       7       1       1       3       4
    5       6       7       8       9      10      11      12      13      14
   15       2       3       6       7      14      15       1       3       3
    1       6       7       1       2       3       1       1       3      19
   28      24      22       2       1       3
```

PROST Array

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

APPENDIX E

Summary of Results

```
BIVAR      PROB       RUE        RISK       CUMRISK
=====      ====       ====       ====       =======

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

SUBSET CALCULATIONS    5           MAX SIZE SUBSET    16
```

APPENDIX F

FORTRAN Listing

```
C        PROBABILISTIC ANALYSIS OF RISKS
C        ADAPTED FROM PAR4 WRITTEN BY J. W. CRUACH JUNE 1967
C        CONTAINS MODIFICATIONS IN LOGIC SUGGESTED BY CRUACH
C        AND MODIFICATIONS IN READ WRITE FORMATS WRITTEN
C        BY L. M. ARNETT  AUGUST 1969
      COMMON/SAVE/KEV(200),KDAT(200),KSTR(200),NCV(200,32),
     1PROST(3000),KHEX(3000)
      COMMON  ROE(200),NGRP(200),ABSPRO(200),NET(200),NIP(200),
     1NUP(200),NB(200),PMUL(200),CURISK(200),RISK(200)
      COMMON PRCM(3000),NFAM(200),NOW(200)
       REAL NZO
      NIN = 5
      NOUT = 6
      KOP=0
      NZO=1.0E-30
      READ (NIN,1)
      WRITE (NOUT,1)
    1 FORMAT (72H1
     1
      DO 2 I=1,200
      ABSPRO(I)=0.
      NFAM(I)=0
      NUP(I)=0
      NIP(I)=0
      NOW(I) = 0
      KEV(I) = 0
      KDAT(I) = 0
      KSTR(I) = 0
      DO 2  J=1,32
    2 NCV(I,J) = 0
      DO 3 I=1,3000
      PROST(I) = 0.
      PRCM(I) = 0.
    3 KHEX(I) = 0
      READ(NIN,6)NTOT,KOPT,NMAX,KHO
    6 FORMAT(15I5)
      IF (NMAX .EQ. 0) NMAX = 25
      MARK = 0
      DO 10 I=1,NTOT
      READ(NIN,7)NVAR,NFORM,ROE(I),KE,KA,(NCV(I,J),J=1,9)
    7 FORMAT(2I5,E15.5,11I5)
      IF(NCV(I,9).EQ.0) GO TO 200
      IF(KE.LE.9) GO TO 200
      READ(NIN,201) (NCV(I,J),J=10,KE)
  201 FORMAT(16I5)
  200 CONTINUE
      IF(I .NE. NVAR) GO TO 15
      KEV(I) = KE
      KSTR(I) = MARK + 1
      IF(NFORM.EQ.1) GO TO 8
      LENG = 2**KE
      KK1 = MARK+1
      KK2 = MARK+LENG
      READ (NIN,9) (PROST(J),J = KK1,KK2)
    9 FORMAT(16E5.2)
      IA = 0
      DO 4 K=1,LENG
      KK3 = MARK+K
      IF (PROST(KK3) .LT. NZO) GO TO 4
      IA = IA + 1
      KK3 = MARK+IA
      KHEX(KK3) = K-1
      KDAT(I) = IA
      KK4 = MARK+K
      PROST(KK3) = PROST(KK4)
    4 CONTINUE
      LENG = IA
      GO TO 10
    8 KDAT(I) = KA
      KK1 = MARK+1
      KK2 = MARK+KA
      IF(KHD.NE.0) GO TO 225
      READ (NIN,5) (KHEX(J),PROST(J),J = KK1,KK2)
    5 FORMAT (5(I10,E5.2),5X)
      GO TO 221
  225 READ(NIN,222)(KHEX(J),PROST(J),J=KK1,KK2)
  222 FORMAT(5(Z10,E5.2),5X)
  221 LENG=KA
   10 MARK = MARK + LENG
      GO TO 20
   15 WRITE(NOUT,16)
   16 FORMAT (///,30X,17HINPUT ERROR. STOP)
      STOP
C                                 LIST INPUT DATA
   20 WRITE(NOUT,17)
   17 FORMAT (55X,10HINPUT DATA/55X10H==========)
      WRITE(NOUT,18)
   18 FORMAT (//22X,46HVAR       NUMBER OF CONTINGENTS    DATA ENTRIES/
     122X46H===      ====================    ===========/)
      DO 19 I=1,NTOT
   19 WRITE(NOUT,14) I,KEV(I),KDAT(I)
   14 FORMAT(22XI3,16XI3,17XI5)
      IF(KOPT.EQ.0) GO TO 21
      IF(LGLAND(KOPT,1).EQ.0) GO TO 130
      WRITE(NOUT,120)
  120 FORMAT (//,55X,9HNCV ARRAY/55X,9H=========/)
      DO 122 I=1,NTOT
      KE = KEV(I)
  122 WRITE(NOUT,123) I,(NCV(I,J),J=1,KE)
  123 FORMAT(45XI3,10X16I4,/58X16I4)
  130 IF(LGLAND(KOPT,2).EQ.0) GO TO 21
      WRITE(NOUT,131)
  131 FORMAT (//,55X,10HKSTR ARRAY/55X,10H==========/)
      WRITE(NOUT,132)(I,KSTR(I),I=1,NTOT)
  132 FORMAT(5(10XI3,I10))
      WRITE(NOUT,133)
  133 FORMAT (//,55X,10HKHEX ARRAY/55X,10H==========/)
      WRITE(NOUT,134) (KHEX(J),J=1,MARK)
  134 FORMAT (5X,10I12)
      IF(LGLAND(KOPT,4).EQ.0) GO TO 21
      WRITE(NOUT,143)
  143 FORMAT (//,55X,11HPROST ARRAY/55X,11H===========/)
      WRITE(NOUT,144) (PROST(J),J=1,MARK)
  144 FORMAT(5X,10E12.3)
C                                  INITIATE SUBSET COMPUTATION
   21 CONTINUE
      NCALC=0
      MAXSUB=1
      JUP=NTOT
  600 DO 601 I=1,JUP
      J=JUP-I+1
      IF(NUP(J).NE.0) GO TO 601
      NFAM(J) =1
      GO TO 603
  601 CONTINUE
      GO TO 25
  603 JUP = J
      CALL FAM(JUP,NIP)
      DO 604 I=1,NTOT
      IF(NIP(I).EQ.1) NUP(I)=1
  604 CONTINUE
      GO TO 600
C                NFAM=1 MARKS FAMILY GENERATORS
   25 NLO = 10000
      IT=NTOT
      LAST=0
      KAN=0
C            SELECT TRIAL SUBSET WITH MAX INDEX KAN
   27 KAN=KAN + 1
      IF(KAN.LE.NTOT) GO TO 30
      IF (NLO.EQ.10000) GO TO 100
      GO TO 55
   30 IF(NOW(KAN) .EQ. 1) GO TO 27
      DO 32 I=1,NTOT
   32 NIP(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
      NCT =0
      NAK=1
      DO 40 I=1,NTOT
      IF (NET(I) .EQ. 1) NCT =NCT+1
      IF (NET(I) .EQ. 1) GO TO 40
      IF (NOW(I) .EQ. 1) GO TO 40
      NUM=KEV(I)
      IF (NUM .EQ. 0) GO TO 40
      DO 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. NLO .AND. LAST .EQ. 0) GO TO 27
      DO 48 I=1,NTOT
      IF (NFAM(I) .NE. 1) GO TO 48
      CALL FAM(I,NUP)
      KILL =0
      DO 41 J=1,NTOT
      IF(NET(J).NE.2) GO TO 41
      IF(NUP(J).NE.1) GO TO 41
      KILL=KILL+1
      IF(KILL .LE. 1)GO TO 42
      NIP(JROD)=2
      NIP(J) = 2
   42 JROD = J
   41 CONTINUE
   48 CONTINUE
      NOK = 0
      DO 50 I =1,NTOT
      IF (NET(I) .NE. 2) GO TO 50
      IF (NIP(I) .EQ. 2) GO TO 50
      NIP(I)=4
      NOK=1
   50 CONTINUE
      IF (NOK .EQ. 1) GO TO 54
      IF (NAK .EQ. 1) GO TO 54
C                SUBSET NOT ACTIONABLE. LOOP AGAIN
      GO TO 27
C     CANDIDATE. COUNT MEMBERS AS NCT
   54 IF(NCT.EQ.1) GO TO 60
      IF(LAST .EQ. 1) GO TO 60
C     SAVE NCT AS NLO IF IT IS LOWEST YET FOUND
      IF(NCT .GE. NLO) GO TO 27
      NLO =NCT
      IT = KAN
      GO TO 27
   55 LAST = 1
      KAN = IT
      GO TO 30
C              SUBSET OF LOWEST INDEX KAN WITH NCT
C              MEMBERS NOW TO BE COMPUTED. MEMBERS ARE GIVEN AS NET EQUAL 1.
C              INSTALL IN NGRP
   60 NGRUP=0
      DO 62 I=1,KAN
      IF(NET(I) .NE.1) GO TO 62
      NGRUP=NGRUP+1
      NGRP(NGRUP) = I
   62 CONTINUE
C           IDENTIFY IN NIP THE FOLLOWING CODES
C                NIP =0 IGNORE
C                NIP =1 UNRESOLVABLE
C                NIP =2 IRREDUCIBLE
C                NIP =3 RESOLVABLE
C                NIP =4 REDUCIBLE
      NO=0
      DO 65 I= 1,NTOT
      IF(NIP(I).EQ.4) GO TO 65
      IF(NET(I).EQ.1) GO TO 65
      IF (NOW(I).EQ.1) GO TO 65
      NUM=KEV(I)
      NIP(I)=0
      IF(NUM.EQ.0) GO TO 65
      DO 66 J=1,NUM
      NX=NCV(I,J)
      IF (NET(NX).NE.1) GO TO 66
      IF(NCV(I,J).GT.NO) NO=NCV(I,J)
   66 CONTINUE
   65 CONTINUE
      DO 67 J=1,NCT
      KA=NGRP(J)
      IF(KA.LE.NO) GO TO 68
      NIP(KA)=3
      GO TO 67
   68 NIP(KA) = 1
   67 CONTINUE
C              CALCULATE SUBSET PROBABILITIES
      IF(NCT .LE. NMAX) GO TO 610
      WRITE (NOUT,612) NCT,NMAX
  612 FORMAT (//34X,14HSUBSET SIZE OF,I3,1X,28HEXCEEDS MAXIMUM SPECIFIED
     1 AS,I3,//34X,30HTERMINATE WITH PARTIAL SUMMARY,///)
      GO TO 100
  610 NCALC = NCALC+1
      IF (NCT .GT. MAXSUB) MAXSUB = NCT
      KUP = NCT
      DO 347 I=1,KUP
      M=NGRP(I)
      NET(M)=0
```

```
347 NB(I) = 0
    K=1
    GO TO 348
349 LA = KUP
365 IF (NB(LA) .EQ. 0)GO TO 367
366 NB(LA) = 0
    M=NGRP(LA)
    NET(M)=0
    LA=LA-1
    IF(LA.LT.1) GO TO 375
    GO TO 365
367 NB(LA) = 1
    M = NGRP(LA)
    NET(M)=1
    NUM = KEV(M)
    JX = KSTR(M)
    J = JX
    IF(NUM.EQ.0) GO TO 370
    L = 0
    DO 368 J = 1,NUM
    LO = NCV(M,J)
    IF (NET(LO) .EQ. 1) L = L+2**(NUM-J)
368 CONTINUE
    JY = JX + KDAT(M) -1
    DO 369 J=JX,JY
    IF(L.EQ.KHEX(J)) GO TO 370
369 CONTINUE
    GO TO 366
370 PMUL(LA) = PROST(J)
    IF(LA.EQ.KUP) GO TO 361
    K = LA + 1
348 DO 360 I=K,KUP
    PMUL(I) = 1.0
    M = NGRP(I)
    NUM = KEV(M)
    JX = KSTR(M)
    J = JX
    IF(NUM.EQ.0) GO TO 355
    L = 0
    DO 350 J=1,NUM
    LO = NCV(M,J)
    IF (NET(LO).EQ. 1) L = L+2**(NUM-J)
350 CONTINUE
    JY = JX + KDAT(M) - 1
    DO 352 J=JX,JY
    IF(L.EQ.KHEX(J)) GO TO 355
352 CONTINUE
    GO TO 360
355 PMUL(I) = 1. - PROST(J)
360 CONTINUE
361 PROB = 1.0
    DO 363 I=1,KUP
363 PROB = PROB*PMUL(I)
C                                 ASSIGN PROBABILITY INCREMENTS
    DO 385 I=1,NTOT
    IF(NIP(I).LE. 1) GO TO 385
    IF(NIP(I).NE. 3) GO TO 371
    IF(NET(I).EQ. 0) GO TO 385
    ABSPRO(I) = ABSPRO(I)+PROB
    GO TO 385
371 KX = KSTR(I)
    KY = KX + KDAT(I) - 1
```

```
      NUM = KEV(I)
      LOOK = 0
      JB = 0
      DO 378 J=1,NUM
      LO = NCV(I,J)
      IF(LO.GT. KAN) GO TO 380
      IF(NIP(LO).EQ. 0) GO TO 378
      JOT = 2**(NUM-J)
      JB=JB+JOT
      IF (NET(LO).EQ. 0) GO TO 378
      LOOK=LOOK + JOT
  378 CONTINUE
  380 DO 382 L=KX,KY
      JA = KHEX(L)
      JC=LGLAND(JA,JB)
      IF(JC.NE.LOOK) GO TO 382
      PRCM(L) = PRCM(L) + PROB
  382 CONTINUE
  385 CONTINUE
      GO TO 349
C                 CONSOLIDATE KHEX AND PROST ARRAYS
C                 RESET KEV, KDAT AND NCV
  375 DO 395 I = 1,NTOT
      IF(NIP(I) .EQ. 3) NOW(I) =1
      IF(NIP(I) .NE. 4) GO TO 395
      JX =KSTR(I)
      JY = JX + KDAT(I) - 1
      NUM = KEV(I)
      DO 391 J = 1,NUM
      LO = NCV(I,J)
      IF(NIP(LO) .NE. 0) NCV(I,J) = 0
  391 CONTINUE
      DO 405 L=JX,JY
      PROST(L) = PROST(L)*PRCM(L)
      PRCM(L)=0.
      JA = KHEX(L)
      JB = 1
      JC = 1
      JD = 0
      DO 410 J=1,NUM
      LA = NUM-J+1
      IF (NCV(I,LA).EQ. 0) GO TO 410
      IF (LGLAND(JA,JC).EQ.0) GO TO 409
      JD=JD+JB
  409 JB=2*JB
  410 JC=2*JC
  405 KHEX(L) = JD
      LA=NUM
      L=0
      DO 411 J=1,NUM
      IF(NCV(I,J) .EQ. 0) GO TO 415
      L=L+1
      NCV(I,L) = NCV(I,J)
      GO TO 411
  415 LA=LA-1
  411 CONTINUE
      KEV(I)=LA
      DO 388 J = JX,JY
      IF (PROST(J) .LT. NZO) GO TO 388
      LA = KHEX(J)
      IF (J .EQ. JY) GO TO 388
      JW = J + 1
      DO 389 JA=JW,JY
      IF (PROST(JA) .LT. NZO) GO TO 389
      IF(KHEX(JA).NE.LA) GO TO 389
      PROST(J) = PROST(J) + PROST(JA)
      PROST(JA) = 0.
      KHEX(JA) = 0
  389 CONTINUE
  388 CONTINUE
      LA = 0
      DO 390 J=JX,JY
      IF (PROST(J) .LT. NZO) GO TO 390
      LA = LA + 1
      KK3 = JX + LA - 1
      PROST(KK3) = PROST(J)
      KHEX(KK3) = KHEX(J)
  390 CONTINUE
      KDAT(I) = LA
      IF(LA.EQ.0) KDAT(I) = 1
  395 CONTINUE
C                                COMPUTATION OF SUBSET COMPLETE
C                                LOOP BACK FOR NEXT SUBSET
      GO TO 25
C                                COMPUTE RISKS AND TOTAL
  100 SUM = 0
      IF(KEV(NTOT).NE.0) GO TO 102
      M=KSTR(NTOT)
      ABSPRO(NTOT)=PROST(M)
  102 DO 103 I=1,NTOT
      RISK(I) = ROE(I)*ABSPRO(I)
      SUM = SUM + RISK(I)
  103 CUMRISK(I) = SUM
C                                WRITE RESULTS
      WRITE(NOUT,105)
  105 FORMAT (//,55X,7HSUMMARY/55X,7H=======)
      WRITE(NOUT,107)
  107 FORMAT(//34X,49HBIVAR        PROB         ROE         RISK       CUMRISK/
     134X,         49H=====        ====         ====        ====       =======/)
      DO 110 I=1,NTOT
  110 WRITE(NOUT,109) I,ABSPRO(I),ROE(I),RISK(I),CUMRISK(I)
  109 FORMAT(33X,I3,4X,E11.4,4X,E8.2,2X,E8.2,3X,E8.2)
      WRITE(NOUT,111)NCALC,MAXSUB
  111 FORMAT (//34X,19HSUBSET CALCULATIONS,I5,10X,15HMAX SIZE SUBSET,I5)
      STOP
      END
```

```
C                   SUBROUTINE FAM FOR PAR4
      SUBROUTINE FAM(KAN,NET)
      COMMON/SAVE/KEV(200),KDAT(200),KSTR(200),NCV(200,32),
     1PROST(3000),KHEX(3000)
      INTEGER NET(200)
      DO 32 I=1,200
   32 NET(I)=0
      NET(KAN)=1
      NUM = KEV(KAN)
      IF(NUM.EQ.0) GO TO 40
      DO 35 J=1,NUM
      KA = NCV(KAN,J)
   35 NET(KA) = 1
      KANDN = KAN - 1
      DO 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
      DO 39 J=1,NUM
      KA = NCV(IA,J)
   39 NET(KA)=1
   38 CONTINUE
   40 RETURN
      END
```