SHARE PROGRAM LIBRARY AGENCY

PROGRAM NUMBER

340001

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257

SHARE PROGRAM LIBRARY SUBMITTAL FORM

SHARE PROGRAM LIBRARY AGENCY
Triangle Universities Computation Center
Post Office Box 12076
Research Triangle Park, North Carolina USA 27709

SPLA CONTROL NUMBER: 237

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the SHARE Reference Manual, Section 6.

(1) Program Number (to be filled by SPLA) 360D-34.0.001

(2) Title of Program . . . . . FLIP: A computer program for fuzzy reasoning

(3) System Type(s) (Machine) . . . . . Used on IBM 360/50 and others

(4) Search Key(s) . . . . . Fuzzy, logic, reasoning

(5) Programming Systems/Languages . . . . FORTRAN

(6) Primary Subject Code . . . . .

(7) Minimum System Requirements

(8) New (N) or Revision (R) (if revision, show prior Program Number in Item 1) N

(9) Date of Submittal . . . . . 2 Aug. 1979

(10) Documentation (number of original pages submitted) . . . . . 1 + 1 + 25 + form

(11) Author's Name and Address . . . . Robin Giles
Mathematics Dept
Queen's University
Kingston, Ont K7L 3N6
CANADA

(12) Direct Technical Inquiries to Name & Address (if different than Author)

(13) Submitter's Installation Membership Code . . . . QK

(14) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

SF X 1001 Revised 11 76

SHARE PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide:

a. Purpose
b. Programming Language used
c. Version and modification level or release number
d. Field of application
e. Type of routine (main program, subroutine, etc.)
f. Specific description of machine requirements

See attached pages

DISCLAIMER

~~Triangle Universities Computation Center (TUCC)~~ serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

(Please attach additional pages if necessary) . . . . . Total pages attached 2

An "Acknowledgement of Assistance" statement must be attached to this Submittal Form.

Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program"

(15) Signature of Submitter and Date R Giles 2 August 1979

(15) Signature of Installation Addressee

SPLA 1001 (R) Revised 11-76

FLIP: A COMPUTER PROGRAM FOR FUZZY REASONING

ABSTRACT

This is an interactive computer program, intended primarily for use at a screen terminal, which implements the procedure proposed in "A formal system for fuzzy reasoning" (Fuzzy Sets and Systems, Vol. 2, No. 3, 1979. The problem in question is that of deciding what conclusions may be drawn in the presence of (possibly conflicting) evidence provided, generally with associated partial degrees of belief, by several sources of differing reliability. In using the program, each piece of evidence is entered as a sentence (using the terms NOT, AND, OR, IMPLIES as necessary), with an associated "degree of belief" and "weight"; followed by a tentative conclusion. The system returns the degree(s) of belief and weight(s) which may rationally be attached to the conclusion.

The program is written in FORTRAN (868 lines, compatible with FORTRAN IV and with FORTRAN V). It has been used on the IBM 360/50. The program is described in "A computer program for fuzzy reasoning" (submitted to Fuzzy Sets and Systems; preprints may be obtained from the author).

Tape Key:

The submitted tape is an IBM standard labelled tape containing one file titled FLIP14. The tape is recorded in 9-track format at a density of 1600 bpi. The IBM utility IEBGENER can be used to punch a deck or print a listing. Control card information would be as follows:

```
//SYSUT1 DD DSN=FLIP14,UNIT=TAPE,
    VOL=SER=123456, LABEL=(1,SL),
    DCB=(RECFM=FB,LRECL=80,BLKSIZE=2400),
    DISP=OLD
```

File Description: Program source deck
sequence numbers in cols 73-80,
00000100 through 00086800
868 card images blocked 30 per block
EBCDIC

FLIP: A COMPUTER PROGRAM FOR FUZZY REASONING

SUPPLEMENTARY NOTES

The purpose and operation of the program is described in "A computer program for fuzzy reasoning" by R. Giles (to appear in Fuzzy Sets and Systems, preprints available from the author). These notes are intended only as a supplement to this paper.

The output of the program is normally written only to the screen, assumed to be logical unit #6. To obtain in addition hardcopy of the output enter "$" and in response to the request "Enter LP, LEV, LSIMP, LPAR, LPTS, LOOP" enter a single integer, the logical unit number assigned to the line printer (free format). This value is then assumed by the variable LP and duplicate WRITE statements are invoked.

Various monitoring facilities used in development and debugging have deliberately been left in the program. They may be invoked, after typing "$", by entering nonzero values for the integers LEV, LSIMP, LPAR, LPTS. This will cause the values of certain variables and arrays to be listed (see the screen response to "$"). The value of LOOP controls the maximum number of iterations allowed in the linear programming routines. The default value of LOOP is 2*DIM(DIM=ITEMS+VARS, where ITEMS = number of items of evidence, and VARS = number of variables = total number of atomic sentences occurring). This value should be sufficient, but if the message, MAXIMUM NUMBER OF ITERATIONS EXCEEDED, is received a higher value may be tried.

Comments and queries concerning the program will be welcomed, and I should be particularly grateful for news of practical problems in which the program has been used, and for suggestions for extensions in the scope of the program that would make it more useful.

R. Giles

Mathematics Department
Queen's University
Kingston
Canada August 2, 1979.

A COMPUTER PROGRAM FOR FUZZY REASONING

Robin Giles

Department of Mathematics and Statistics
Queen's University
Kingston, Ontario, Canada

Queen's Mathematical Preprint No. 1979-17

Abstract

An interactive computer program is described which implements the procedure proposed in "A Formal System for Fuzzy Reasoning" [Fuzzy Sets and Systems, ...]. The problem in question is that of deciding what conclusions may be drawn in the presence of (possibly conflicting) evidence provided, generally with associated partial degrees of belief, by several sources of differing reliability. In using the program, each piece of evidence is entered as a sentence (using the terms NOT, AND, OR, IMPLIES as necessary), with an associated "degree of belief" and "weight"; followed by a tentative conclusion. The system returns the degree(s) of belief and weight(s) which may rationally be attached to the conclusion.

A COMPUTER PROGRAM FOR FUZZY REASONING

1. Introduction

Classical logic allows the logical consequences of a number of given statements to be determined. For instance, if I am informed of the truth of the four statements A or B , A implies C , not B , B implies D it assures me that C is true, that nothing can be inferred regarding the truth or otherwise of D , and so on. However, in practice the situation is more complex: an informant does not generally make an assertion with certainty, but ascribes to it a certain degree of belief, usually by hedges such as "probably", "almost certainly", "possibly", and so on. In addition - and independent of this - I may consider some informants more reliable than others, and wish to assign greater weight to their assertions in the process of arriving at a conclusion on the basis of the information received. A judge, for instance, is in exactly this position when, in court, he has to reach a verdict on the basis of the assertions of various witnesses. In this case a further complication is often present: the assertions offered in evidence are generally to some extent inconsistent with each other. In such a case classical logic fails completely: certainly, it assures the judge that the accused is guilty, but it also assures him of his innocence - in classical logic every statement is a logical consequence of inconsistent evidence! A similar situation arises in many other fields. The chairman of a committee, for instance, faces it

whenever he wishes to formulate some weighted average of the opinions expressed in a discussion. The same problem confronts a doctor who has to reach a diagnosis on the basis of his own observations together with evidence submitted by specialists in various fields - radiology, psychiatry, biochemistry, and so on; or an engineer, who may receive evidence from various sources regarding the cause of trouble in - for instance - a malfunctioning nuclear reactor.

In [1] a formal system was described which allows rational conclusions to be reached in such cases, providing of course that the necessary degrees of belief and weights have been expressed in a suitable quantitative form. The practical implementation of this system in the situation in question involves two stages. In the first stage, every assertion offered in "evidence" is expressed as a formal sentence with an associated <u>degree of belief</u> and <u>weight</u>, and similar formal expression is given to any tentative conclusion on which a ruling is required. There is no reliable way of avoiding the difficulties involved in this process, since it is necessary to represent formally not simply the words used by an informant but rather the <u>meaning behind the words</u>. As was shown in discussing a concrete example in [1], to do this conscientiously may necessitate an interview with the informant to elicit his exact meaning.

In the second stage a calculation is carried out which yields the[1] degree of belief b and weight w that may rationally be attached to the conclusion. This calculation[2] is nontrivial and

---

1 In the general case several such pairs (b,w) may result. [One may be justified in asserting a moderate degree of belief with high reliability (large weight), or a greater degree of belief with less reliability.]

2 An example is treated in detail in [1].

---

sufficiently tedious to deter a would-be user from trying out the system in practice. FLIP (for Fuzzy Logic Interactive Program) is a computer program which carries out all the work involved in this second stage. To facilitate its use FLIP is written in standard Fortran IV and can be used in an interactive mode on a microcomputer[3] with a display screen terminal. No preparation of

---

3 It has been operated on Digital Equipment Corporation's PDP11/V03.

---

the data is necessary; the sentences offered in evidence are simply typed in, with their weights and degrees of belief, in response to requests from the system, followed by the proposed conclusion terminated by "?". FLIP then returns the degree(s) of belief and weight(s) which may be ascribed to the conclusion. As an example, I give (fig. 1) the display for the problem worked in [1]: Granny has said that young John probably has a fever (degree of belief .7), John says he has not (degree of belief 1),

and John's doctor says (on the phone) that if he has a fever the odds are 4 to 1 (i.e. degree of belief .8) that he should go to bed. The weights assigned to these assertions are 4, 1, and 10 respectively. On receiving the query, BED? (an abbreviation for "John should go to bed"), FLIP remarks on the inconsistency in the evidence (there is a clash between the assertions of Granny and John; for the numerical measure of inconsistency see Sec.2) and then reports the weight and degree of belief that may be assigned to the queried sentence. (The run of FLIP is left open. It will be continued in Fig. 6.)

```
  ENTER FIRST ITEM.
FEVER, BLF .7, WT 4
  ENTER NEXT ITEM, OR "BYE" TO STOP.
NOT FEVER
  ENTER NEXT ITEM, OR "BYE" TO STOP.
FEVER IMPLIES BED, BLF .8, WT 10
  ENTER NEXT ITEM, OR "BYE" TO STOP.
BED?
  THE EVIDENCE SHOWS INCONSISTENCY OF WEIGHT   0.70 .
  FOR THE SENTENCE QUERIED A DEGREE OF BELIEF 0.50
  MAY BE ASSERTED WITH WEIGHT   3.00 .
  ENTER NEXT ITEM, OR "BYE" TO STOP.
```

Fig. 1

Since no effective use of the program can be made without handling the first stage mentioned above I give in Sections 2 and 3 an outline of the logical system described in [1]. This is necessary to understand (a) the way in which degrees of belief and weights are assigned and (b) the meanings attached to the logical connectives used in this work. More detailed information about the program is given in Section 4.

2. The logic

The logical system on which FLIP is based depends on the principle that any assertion is represented by a commitment. In the case of a "simple sentence", i.e. one not containing the logical terms "and", "or", or "implies" (but possibly containing "not"), the commitment takes the form of a bet. The odds offered represent the degree of belief, the stake (i.e. the sum wagered) corresponds to the weight attached to the assertion, and the circumstances under which the bet becomes payable are determined by the sentence asserted. To take a simple example, suppose my friend Bill asserts "It will probably rain today". The sentence involved is "It will rain today", which we may denote by the Fortran symbolic name "RAIN".[4] The hedge, "probably",

---

4 Note that this sentence is such that it is possible to agree on a procedure by which it might - albeit at some time in the future - be tested, or, more precisely, on a procedure by which a bet involving the sentence should be settled. The present approach is fundamentally pragmatic in that it admits only sentences of this type.

---

indicates an associated degree of belief. To assess it quantitatively we need Bill's guidance. Suppose he agrees that his use of "probably" may be interpreted as meaning that the subjective probability he attaches to the sentence is at least 0.7 . Then we assign to RAIN a degree of belief .7 , meaning by

this that his statement is now represented by an offer to bet on the occurrence of rain at odds of 7 to 3 $(.7 \div (1 - .7))$.

With only one assertion the assignment of weight is arbitrary. Let us suppose, however, that my neighbour Jim also expresses an opinion. He says, "The chance of rain today is at least 40%". (He is an amateur meteorologist!) In view of its precision we can record his assertion at once: "RAIN, BLF .4". Weights are used to represent the relative reliability of the two informants. Suppose that, in view of Jim's special interests, we decide to assign a weight of 2 to his assertion, and 1 to Bill's. [The principle used here is that (in a given context) a speaker should be assigned weight n if an assertion of his is considered as being neither more nor less significant than the receiving of n similar assertions from n independent speakers of weight 1.] Jim's assertion may now be fed to FLIP as "RAIN, BLF .4, WT 2", and Bill's as "RAIN, BLF .7". (The default values of WT and BLF are 1.)

Insofar as the logic is concerned the significance of the weights lies in the way the assertions are represented as bets: Jim's assertion is represented as an offer to bet (up to) \$2, while the upper limit on Bill's stake is only \$1. As we shall see, this distinction has the desired effect of attaching twice as much importance to Jim's assertion.

For computational purposes we can represent Jim's assertion by its risk function. This is a function f, defined on the interval [0,1], such that f(x) is the average value of the gain I might make (in virtue of Jim's bet) if I knew that the

probability of rain was $1-x$. Thus here $f(x) = \max\{0, 2(.4-x)\}$. Indeed, if $x > .6$ then the probability of rain is less than .4 so it pays me to accept Jim's bet - I make an average net gain of $\$2(x-.6)$, while if $x \leq .6$ I would not accept it, so my gain would be zero. The function f is called a <u>risk</u> function since $f(x)$, the <u>risk value</u> of Jim's assertion, represents (my estimate of) the risk incurred by Jim in offering the bet. [It is technically more convenient to take (as here) x as the probability of not-rain rather than as the probability of rain: x itself is then just the risk value of the simple sentence RAIN .]

Similarly, Bill's assertion is represented by the risk function $g(x) = \max\{0, x-.3\}$

The net value to me of the two assertions is now given by $e(x) = f(x) + g(x)$, which is graphed as the solid line in fig. 2; we call e the <u>risk function of the evidence</u>.

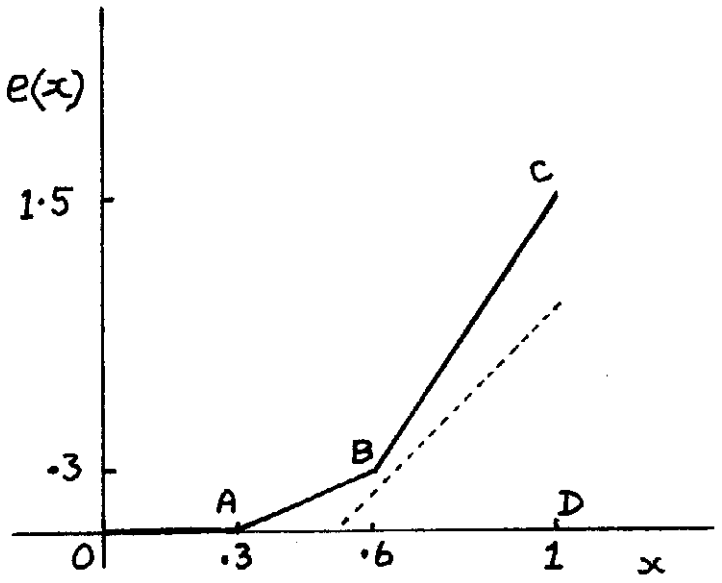

Fig. 2

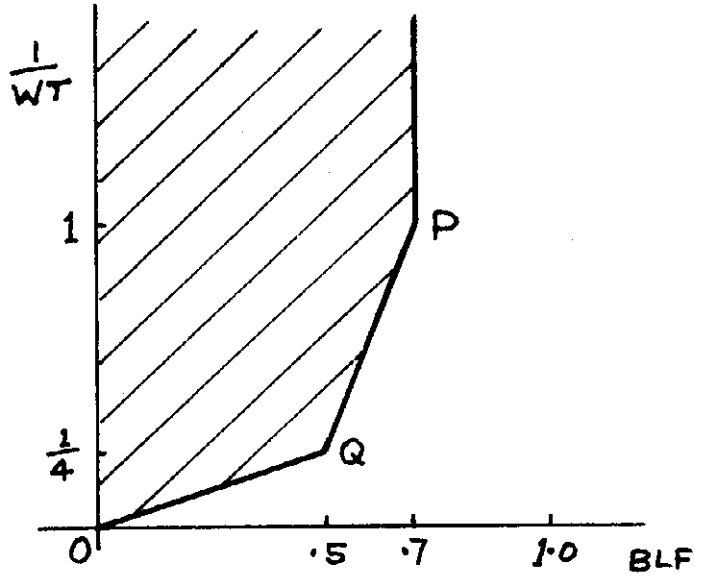

Fig. 3

Now, from this evidence, what can I safely conclude about rain? A reasonable way of answering this question is to describe what assertions about rain I can safely make. For example, I can safely assert "RAIN, BLF .5, WT 2"; in fact, the graph of the risk function for this assertion (dotted line in fig. 2) lies below the solid line, which shows that my "income" from the evidence will, whatever the value of x, cover my "expenditure" in fulfilling the commitment represented by this assertion.[5]

---

5 It might be doubted whether this argument is really valid: it only shows that I am financially safe provided I know the "actual" probability x, but in practice I cannot know this. In a forthcoming paper it will be shown that, providing the risk functions involved are "convex" (which means that their graphs (cf. fig. 1) are convex down), the conclusion of the argument is nevertheless valid: I can safely make an assertion iff the graph of its risk function lies below that of the evidence.

Each line that lies below the graph ABC of $e(x)$ (and intersects AD) represents in this way a safe assertion. Among these assertions are two "maximal" ones, corresponding to the lines AB and BC: namely, the assertions "RAIN, BLF .7, WT 1" and "RAIN, BLF .5, WT 3". These are represented by the points P,Q in fig. 3, and it is not difficult to show that the points representing the other "safe" assertions exactly fill the (shaded) convex region overlying the lines OQ, QP. Thus to know all that can be said about RAIN it is enough to know the two maximal assertions; this

information, as we shall see shortly, is provided by FLIP. It should be noted that, although there are lots of "safe" assertions, all these assertions are safe only in the sense that the potential gain arising from the bets offered in evidence is sufficient to cover the risk incurred in uttering any <u>one</u> of them. A safe assertion is not one that involves no risk, but one for which the risk is covered by the expected gain arising from the evidence.

In the above example, the two informants were in agreement that rain was possible. To see what happens in the case of conflicting evidence let us suppose that Bill's wife Joan now asserts "It is not going to rain". If her assertion is assigned weight 1 it can be recorded simply as "NOT RAIN" and has the risk function $h(x) = 1-x$. Adding it to the previous evidence we now get for the risk function $e(x)$ of the total evidence the graph in fig. 4.

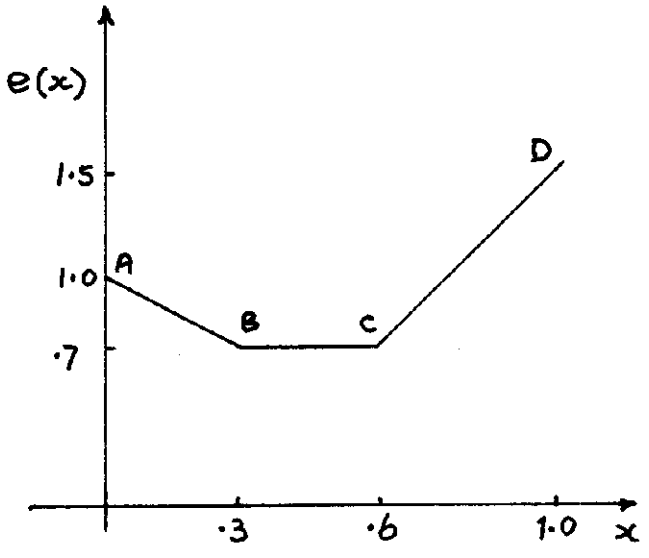

Fig. 4

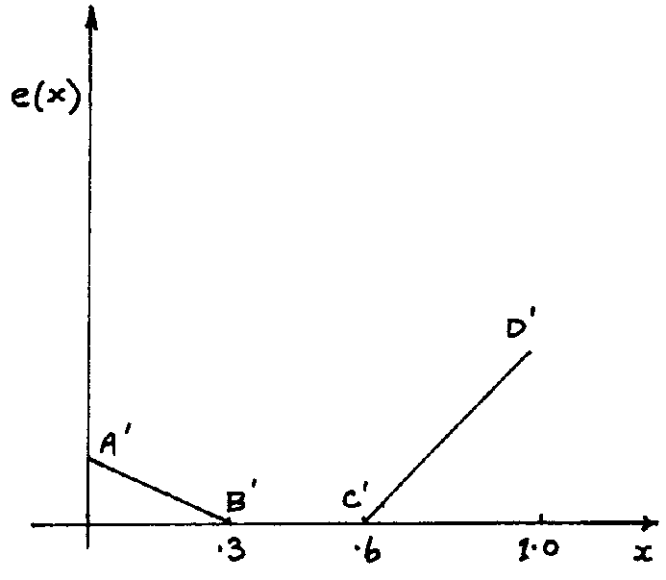

Fig. 5

The conflict between the speakers is now apparent: no value of x is compatible with all three assertions; someone is bound to lose. In fact, whatever the value of x, I have (in virtue of the three bets) an expected gain of at least \$.7;[6] we call .7 the inconsistency of the evidence.

---

6 If I know the value of x I can by suitable betting, ensure an expected gain of at least \$.7. In fact it can be shown that, in the case of a convex function e, I can even without this knowledge bet in such a way as to ensure an expected gain of \$.7. This is true not only in this particular example but in general.

What conclusion can be drawn in the case of conflicting evidence? If we take the same approach as before and look for assertions which can (in the presence of the evidence) be made with no net loss the answer is not very satisfactory. For example, any assertion with a weight of .7 is allowed - even one totally unrelated to the evidence. This is reminiscent of the situation in classical logic that from a contradiction anything follows. It was suggested in [1] that a more satisfactory procedure is to admit as a reasonable inference from the evidence any assertion which can safely be made after renouncing any gain from the evidence which is guaranteed in virtue of its inconsistency: i.e. one uses the same method as before after subtracting the inconsistency from the risk function for the evidence - in effect, contradictory pieces of evidence are allowed to cancel out, and only the balance of evidence after this has been done is used as a

basis for conclusions. The procedure is somewhat ad hoc, but it has at least the virtue that the resulting "reasonable inferences" are now mutually consistent.

As an example, in the present case the risk function of the evidence is first reduced by .7 giving the graph shown in fig. 5. Then the safe assertions are determined. For RAIN there is only one maximal assertion, RAIN, BLF .4, WT 2 , given by the line C'D'. (This of course is a reflection of Jim's evidence, Bill's assertion now having been neutralized by Joan's.) But there is also a maximal assertion for NOT RAIN, corresponding to the line A'B': namely, NOT RAIN, BLF .3, WT 1 .[7]

---

7 At first sight it might seem that these two assertions were inconsistent, since one asserts RAIN and the other NOT RAIN. But this is not so: the first effectively asserts that the probability of rain is $\geq .4$ and the second that it is $\leq .7$.

Fig. 6 shows how these results are obtained using FLIP. Starting with the situation at the end of fig. 1, CLEAR is first typed, resulting in the erasing of the variables FEVER and BED and of the data concerning them. Then Jim's and Bill's evidence is entered and "RAIN?" elicits the two maximal consequences given earlier. Next Joan's assertion is entered (Jim's and Bill's are still in store) after which "RAIN?" produces the single outcome BLF .4, WT 2 , and "NOT RAIN?" yields BLF .3, WT 1 .

```
CLEAR
  ENTER FIRST ITEM.
RAIN, BLF .4, WT 2
  ENTER NEXT ITEM, OR "BYE" TO STOP.
RAIN, BLF .7
  ENTER NEXT ITEM, OR "BYE" TO STOP.
RAIN?
  THE EVIDENCE IS CONSISTENT.
  FOR THE SENTENCE QUERIED ANY ONE OF THE FOLLOWING ASSERTIONS
  MAY BE MADE:
  (1) A DEGREE OF BELIEF 0.70 MAY BE ASSERTED WITH WEIGHT    1.00
  (2) A DEGREE OF BELIEF 0.50 MAY BE ASSERTED WITH WEIGHT    3.00
  ENTER NEXT ITEM, OR "BYE" TO STOP.
NOT RAIN
  ENTER NEXT ITEM, OR "BYE" TO STOP.
RAIN?
  THE EVIDENCE SHOWS INCONSISTENCY OF WEIGHT    0.70
  FOR THE SENTENCE QUERIED A DEGREE OF BELIEF 0.40
  MAY BE ASSERTED WITH WEIGHT    2.00
  ENTER NEXT ITEM, OR "BYE" TO STOP.
NOT RAIN?
  FOR THE SENTENCE QUERIED A DEGREE OF BELIEF 0.30
  MAY BE ASSERTED WITH WEIGHT    1.00
  ENTER NEXT ITEM, OR "BYE" TO STOP.
BYE
```

Fig. 6

3. Compound sentences

It should now be clear how the computer interprets simple sentences, i.e. those that contain no logical terms except perhaps "not", and the significance it attaches to degree of belief and to weight: every sentences is represented by some commitment, which for a simple sentence is simply a bet. Now take a sentence of the form A IMPLIES B , where A and B are simple sentences. This is interpreted as an offer (commitment) by the speaker to bet \$1 on B provided his opponent will bet \$1 on A . As before, we can represent this commitment by a risk function f , but now it is a function of two variables. In fact, suppose x and y are

the risk values (= subjective probabilities of failure) I ascribe to A and B . Then I would accept the speaker's offer if $x < y$, since in this case I would expect to gain on average \$y and lose only \$x , a net gain of \$(y-x) ; on the other hand, if $x > y$ I would not accept the offer. Thus my expected gain is $f(x,y) = \max\{0,y-x\}$, which gives the appropriate risk function.

In practice one may wish to express a certain degree of belief in " A IMPLIES B ". The clue to how this is done is given by the case of a simple sentence. For me to assert " A , BLF .8" is equivalent to saying that I would assert A if you paid me \$.2, for the risk function is in each case $\max\{0,x-.2\}$. So we interpret the assertion, A IMPLIES B , BLF .8 as meaning that the speaker would assert A IMPLIES B if paid \$.2; its risk function is therefore $\max\{0,\max\{0,y-x\}-.2\} = \max\{0,y-x-.2\}$. (A speaker who asserts it expresses a belief that the probability of B's failing is at most .2 greater than that of A's failing.) Degrees of belief for other compound sentences are dealt with in the same way: if P is any sentence, with risk value v , then the risk value of P, BLF b (with $0 \leq b \leq 1$) is $\max\{0,v-(1-b)\}$.

An assertion may also be qualified by an expression of the form "WT w" ($w > 0$). As indicated in Section 2, this modification is dealt with by FLIP by simply multiplying the computed risk function by w , the change being always effected after any value of BLF has been incorporated.

In Section 2 there was one "risk variable" (x), and in this section we have used two (x,y); FLIP is designed to handle up to 10, and can easily be modified to deal with a larger number. In

the case of a problem with n risk variables $x_1,\ldots,x_n$ each risk function will be regarded as a function of the n-tuple $(x_1,\ldots,x_n)$, thus having the domain $[0,1]^n$ (although in practice risk functions will rarely depend on more than 2 or 3 of the x's).

The risk functions which have arisen in this section and in the last have all been of the form

$$f = \max\{0,f_1\}, \tag{1}$$

where f_1 is a (inhomogeneous) linear function of the risk variables. In fact, FLIP is designed to treat only sentences of the type (1) - let us call them sentences of simple max type, avoiding the much more complicated forms - involving both the operations max and min and an arbitrary number of linear functions $f_1,\ldots,f_n$ - which arise in the general case. The reasons for this restriction (which as we shall see is not as serious as it seems) are as follows.

First, as explained in [1], the mathematical problem of drawing conclusions from pieces of evidence rapidly becomes intractible in the general case, but reduces to a problem in linear programming if max operators only are admitted. This means allowing only sentences whose risk functions are of general max type: i.e. of the form $f = \max\{f_1,\ldots,f_n\}$, where $f_1,\ldots,f_n$ are linear. It would be quite practicable to write a program which deals with risk functions of this form, but it is certainly easier - and as it turns out still quite adequate - to treat only the simpler form (1). The risk function f can then be stored as the vector of coefficients of the single linear form f_1 (let

us call this a risk vector). Since we shall normally be dealing with this case we shall abbreviate "simple max type" to "max type" from now on.

With these thoughts in mind let us consider the interpretation of AND and OR . Ever since the original work of Lukasiewicz, it has been customary to represent these connnectives by the operations max and min on truth values (or, equivalently, min and max on risk values); certainly, this is almost always done in current work in fuzzy set theory. This procedure would conflict with the above policy; however, there are other forms for these connectives, called bold conjunction and bold disjunction in [1], which have the same classical limits ([1] fig. 2) and are in fact (see especially [1] Sec. 10) more appropriate in this bet-theoretic context. We shall adopt these as the interpretations of AND and OR . To be specific, this means the following: " A OR B " is first taken as a synonym for either of the equivalent[8] sentences, " (NOT A) IMPLIES B " and

---

8 These are equivalent not merely in the classical but also in the bet-theoretic sense: i.e. they have the same risk function and represent the same commitment.

" (NOT B) IMPLIES A " (the intuitive acceptability of this definition is obvious) and then " A AND B " is defined to mean "NOT((NOT A) OR (NOT B))". With these definitions we get, for the risk values of A OR B and A AND B , $\max\{0,1-x-y\}$ and $\min\{1,x+y\}$, respectively. Of course, A AND B is not of max

type; it is rather of (simple) min type, by which we shall mean that its risk function has the form

(2) $$f = \min\{1, f_1\} ,$$

where f_1 is a (inhomogeneous) linear function of the risk variables.

We have obtained expressions giving the risk values of A IMPLIES B , A OR B , and A AND B , in terms of the risk values x and y of simple sentences A and B ; it is clear too that the risk value of NOT A is 1-x . These same expressions are now taken to apply also when A and B are arbitrary sentences. (This procedure can be justified by a dialogue interpretation of the logical connectives. See for instance [1].) With these definitions every sentence becomes represented by a risk function. For example, take the sentence (A AND B) IMPLIES C , which we will denote briefly as P . If A,B,C have risk values x,y,z then A AND B has risk value $\min\{1,x+y\}$ so that the risk value of P is $\max\{0,z-\min\{1,x+y\}\} = \max\{0,\max\{z-1,z-x-y\}\} = \max\{0,z-x-y\}$.

Note that P is a sentence of max type. The existence of a large class of sentences of max type is given by the following result, which comes immediately from the form of the expressions for the risk functions associated with IMPLIES, OR, AND, and NOT:

Theorem.

(1) Every simple sentence is of both max and min types.

(2) If sentences P and Q are of max type and R and S are of min type then:

(a) P OR Q , NOT R , and R IMPLIES P are of max type, and

(b) R AND S and NOT P are of min type.

Many (but not all) sentences of max type can be seen to be so by means of this theorem. We shall call such sentences admissible sentences; only these sentences are accepted by FLIP, others being rejected with the comment "SYNTAX ERROR" and an indication of the nature of the error. The following are some of the admissible sentences that contain at most three occurrences of simple sentences. [A, B, C denote simple sentences, which need not be distinct. Remember that any or all of A, B, C may contain "NOT" .]

A

A OR B = (NOT A) IMPLIES B = (NOT B) IMPLIES A

A IMPLIES B = B OR (NOT A)

NOT (A AND B) = (NOT A) OR (NOT B) = A IMPLIES NOT B =
B IMPLIES NOT A

A OR B OR C

(A AND B) IMPLIES C = C OR NOT(A AND B) =
A IMPLIES (B IMPLIES C) = C OR (NOT A) OR (NOT B)

A IMPLIES (B OR C) = B OR C OR NOT A

A IMPLIES NOT(B AND C) = NOT (A AND B AND C) =
(NOT A) OR (NOT B) OR (NOT C)

Forms connected by "=" are equivalent: i.e. they are represented by the same risk function. Other equivalent forms

may be obtained by replacing an expression in parentheses by any equivalent expression, using the above equations, the equation NOT NOT A = A , and the fact that AND and OR are associative and commutative (e.g. P AND Q = Q AND P and (P AND Q) AND R

P AND (Q AND R)).

4. Practical use of FLIP

FLIP is designed to solve the problem described in Section 1. Several pieces of evidence have been provided by a number of informants and it is asked whether, and to what extent, a proposed conclusion follows from this evidence.

The first step in invoking FLIP is to identify the "atomic sentences" which occur in the evidence. An atomic sentence is one that contains no logical terms, not even NOT; FLIP can handle up to 10 such sentences. Each such sentence is arbitrarily given a name, which must be a "Fortran symbolic name" (a string of letters and digits, starting with a letter). The name may be of any length, but FLIP will only read the first 6 symbols. We shall refer to these names as variables.

Next, each assertion of an informant must be expressed as a formal sentence using only the terms NOT, AND, OR, and IMPLIES, in addition to variables and parentheses, "(" and ")". (We have seen examples of such sentences in Fig. 1 and Fig. 6.) The sentence may simply be written down directly as a paraphrase of what was said by the informant, but greater reliability will be obtained by

considering whether the informant would be willing to accept the commitment associated (as described in Sec. 3) with the proposed sentence.

In any case, the sentence chosen must be admissible in the sense explained in Section 3. In most cases this is no problem. For example, if RAIN is a variable then both RAIN and NOT RAIN are admissible; and if WIND is another variable then RAIN OR WIND and RAIN IMPLIES WIND are admissible. The conjunction WIND AND RAIN , on the other hand, is not admissible. However, this difficulty can be avoided by instead entering WIND and RAIN as two separate assertions. Similarly, WIND IMPLIES (COLD OR RAIN) is admissible, and WIND IMPLIES (COLD AND RAIN) , which is not admissible, can be replaced by the two assertions WIND IMPLIES COLD and WIND IMPLIES RAIN ; also, (WIND AND RAIN) IMPLIES COLD is admissible, while the inadmissible (WIND OR RAIN) IMPLIES COLD can be replaced by the assertions WIND IMPLIES COLD and RAIN IMPLIES COLD ; and similarly WIND OR (RAIN AND COLD) can be replaced by the pair WIND OR RAIN and WIND OR COLD .

Of course, in all this, OR is the inclusive "or": if, in asserting SUNNY OR RAIN , one wishes to imply "and not both" it is necessary to add, as another assertion, NOT (SUNNY AND RAIN) .

When a sentence, chosen to represent the assertion of an informant, is fed to FLIP one has the option of qualifying it by " BLF b " and/or " WT w ", where b and w are decimal numbers with $0 \leq b \leq 1$ and $0 \leq w$. [Commas may be used freely; they are ignored by FLIP.] b denotes the <u>degree of belief</u> of the

assertion: thus b = 1 denotes certainty and b = 0 denotes complete uncertainty. The appropriate value of b may be chosen intuitively, or - for greater reliability - one may use the fact that $(1-b) is interpreted by FLIP as the sum the informant would have to be paid before he would agree to accept the commitment associated with the sentence asserted, this commitment being determined in the manner described in Section 3. w denotes the weight attached by the listener to the assertion, and is allocated in the manner indicated in Section 2. Default values of b and w are 1.

Up to 10 items of evidence may be entered in this manner. FLIP first reads each item, identifying punctuation marks, numbers, and words. Each word is compared with the entries in a "dictionary" which initially contains only the logical terms and a few commands; if a word is not listed it is assumed to be a variable and is added to the dictionary. The variables are thus numbered in order of their appearance. At the same time as this is going on, a computation of the risk function is carried out, using the recursive procedure explained in Section 3 and at the end incorporating any assigned values of BLF and WT. This involves the calculation of intermediate risk functions, each of which is stored temporarily in the form of a "risk vector" (see Sec. 3). The calculation is thus possible only if the intermediate results are all of simple (max or min) type: i.e. if the asserted sentence is admissible. If not, or in case of some other error in syntax, the whole item is discarded (this fact, and the reason for it, being indicated) and the next item

is requested.

After the items of evidence are fed to FLIP, the corresponding risk vectors being stored as the rows of a matrix EVIDNZ, a tentative conclusion is entered. This is done in exactly the same way as for an item of evidence, except that no weight or degree of belief is entered and the item is terminated by "?". (See figs. 1 and 6 for examples.) The corresponding risk vector is stored by FLIP as RISK, and EVIDNZ and RISK are passed to a subroutine which formulates and solves a linear programming problem (see Appendix) and reports (a) the weight of inconsistency (if any) of the evidence, and (b) the weights and degrees of belief of the (one or more) corresponding maximal assertions. RISK is then discarded, but EVIDNZ is retained so that enquiry may be made (as in fig. 6) of other tentative conclusions, either before or after entering further items of evidence.

At any time, entry of the word "CLEAR" restores FLIP to its initial state (annulling all previously entered items) and entering "BYE" terminates the session.

The program consists of some 800 lines of Fortran. It is intended to make source copies available to users through the program libraries, SHARE, DECUS, and CUBE.

Appendix: Reduction to a linear programming problem

Suppose that there are n risk variables, $x_1,\ldots,x_n$, and m items of evidence $E_1,\ldots,E_m$ are supplied, each being an

admissible sentence. Then the risk function F_i of the ith item has the form $F_i(x_1,\ldots,x_n) = \max\{0, e_{i0} + \sum_{j=1}^{n} e_{ij}x_j\}$. (The matrix $[e_{ij}]$ is the array EVIDNZ, mentioned in Section 4.) Suppose that a tentative conclusion C is mooted, which has risk function $\max\{0, c_0 + \sum_{j=1}^{n} c_j x_j\}$. We wish to know for what values of degree of belief b and weight w this conclusion may safely be asserted. If the evidence is consistent, this amounts to asking for what values of b and w do we have

$$\sum_{i=1}^{m} \max\{0, e_{i0} + \sum_{j=1}^{n} e_{ij}x_j\} \geq \max\{0, w(c_0 + \sum_{j=1}^{n} c_j x_j - 1 + b)\},$$

whenever

(1) $\qquad 0 \leq x_j \leq 1 \quad (0 \leq j \leq n)$.

Clearly the 0 on the right hand side may be ignored, so to answer this question it suffices to solve the following problem:

(2) Find, for each given $w \geq 0$ and with the x's restricted by (1), the minimum value, q_{min}, of

$$q = \sum_{i=1}^{m} \max\{0, e_{i0} + \sum_{j=1}^{n} e_{ij}x_j\} - w(c_0 + \sum_{j=1}^{n} c_j x_j).$$

In fact, if $w = 0$ then this q_{min} is just the inconsistency, INC, of the evidence, while if $w \geq 0$ then a value of b is allowed exactly if $q_{min} - \text{INC} \geq -w(1-b)$. (We have subtracted the value of INC from the risk value of the evidence in accordance with the procedure explained in Section 2.) Thus the maximum value of b (for a given w) is

$$b_{max} = 1 - (INC - q_{min})/w .$$

Now, the above problem is equivalent to the following:

For each $w \geq 0$ minimize

$$q = \sum_{i=1}^{m} y_i - w(c_0 + \sum_{j=1}^{n} c_j x_j)$$

subject to

$$(3) \qquad y_i \geq e_{i0} + \sum_{j=1}^{n} e_{ij} x_j \qquad (i=1,\ldots,m),$$

$$y_i \geq 0 \qquad (i=1,\ldots,m),$$

and

$$0 \leq x_j \leq 1 \qquad (j=1,\ldots,n),$$

for clearly a minimum will be attained only when

$$y_i = \max\{0, e_{i0} + \sum_{j=1}^{n} e_{ij} x_j\} \qquad (i=1,\ldots,m) .$$

Finally, introducing slack variables v_i and z_j this becomes:

For each $w \geq 0$ minimize

$$q = \sum_{i=1}^{m} y_i - w(c_0 + \sum_{j=1}^{n} c_j x_j)$$

$$(4) \quad \text{subject to}$$

$$1 = x_j + z_j \qquad (j=1,\ldots,n),$$

$$-e_{i0} = \sum_{j=1}^{n} e_{ij} x_j - y_i + v_i \qquad (i=1,\ldots,m),$$

with all x's , y's , z's , and v's nonnegative.

This is a standard problem in parametric linear programming, w being the parameter. The method of solution is clearly

explained (and illustrated with examples) in [2]; to help in understanding the program, the procedure described there is followed closely by FLIP.

First the simplex method is used to find a basic feasible solution (bfs) which minimizes q in the case $w = 0$. An initial bfs is always available: in fact, if $e_{i0} \leq 0$ for all i then one may take the z's and the v's as basic variables; and if not we still get a bfs by choosing (for inclusion in the basis) y_i instead of v_i whenever $e_{i0} > 0$. Using this basis, an array TABLO - which is a simplex tableau (in the "compact form", [2] p. 40) for the equations (4) - is set up by FLIP, and the usual iterative pivot procedure is employed to find an optimal bfs S_0. Then the parametric programming procedure of [2] pp. 131-145 is implemented. This generates a finite sequence of bfs's, $S_1, S_2, \ldots, S_N$. The solution S_0 is optimal for w-values in an interval $[0,w_1]$, S_1 is optimal for w in $[w_1,w_2]$, and so on, where $0 \leq w_1 \leq w_2 \leq \ldots \leq w_N = \infty$. To determine the "maximal assertions" (see Sec. 3) that can be made concerning the queried sentence it is not necessary to record, for each bfs, the values of all variables. It is enough to note the values of the two quantities $c_0 + \Sigma c_j x_j$ and Σy_i. These are recorded by FLIP as XCOORD(I) and YCOORD(I) (I=1,N); they correspond to the X and Y coordinates of the points occuring in a generalization of the diagrams of Fig. 2 and Fig 4. It is then a simple matter (see the examples in Section 2 and also [1] Sec. 5.6) to determine from this sequence of points the values of BLF and WT corresponding to the "maximal assertions" (Sec.

2). These values are reported by FLIP; from them all the safe assertions can be determined easily as shown in Section 2.

References

[1] R. Giles, A formal system for fuzzy reasoning, to appear in Fuzzy Sets and Systems.

[2] C. Van De Panne, Methods for linear and quadratic programming (North-Holland, Amsterdam, 1975).