PROPA Version 1.6 - A Probabilistic Performance Analysis Program for Availability Simulation of 3-State Systems

Yoichi Watanabe

November 1985

UWFDM-657

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

PROPA Version 1.6 - A Probabilistic Performance Analysis Program for Availability Simulation of 3-State Systems

Yoichi Watanabe

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

November 1985

UWFDM-657

PROPA Version 1.6 -

A *Pro*babilistic *P*erformance *A*nalysis Program for Availability Simulation of 3-State Systems

Yoichi Watanabe

Fusion Technology Institute
The University of Wisconsin
1500 Johnson Drive
Madison, WI 53706

November 1985
revised December 1986

UWFDM-657

Contents

List of Figures

1 Introduction

A Monte Carlo simulation methodology for 3-state systems with 3-state components has been developed and a computer program PROPA (PRObabilistic Performance Analysis) has been written[1]. The program was used to analyze the STARFIRE tokamak fusion reactor plant availability[2].

The basics of the computational methods used in the program were discussed in Ref.[1]. In this report we shall describe precise models utilized in the PROPA program(Version 1.6). Following the description of the models in Section 2, descriptions of input and output will be given in Section 3. Section 4 will discuss the program structure in detail.

2 Model

2.1 System Model

- A system consists of components, which are not further decomposed.
- All elements (the system, subsystems, and components) assume three states: normal, degraded, and failed. These three states are denoted by 2, 1, and 0 in that order.
- An element is associated with only one type of state quantity.
- Components can be dependent on one another.
- The states of the entire system and subsystems are obtained by means of a systems tree from states of components and other subsystems and gate logic in the systems tree.

2.2 Failure Model

- A component in state 2 fails into either state 1 or 0 with failure rate λ_{21} or λ_{20} as illustrated in Fig.1.

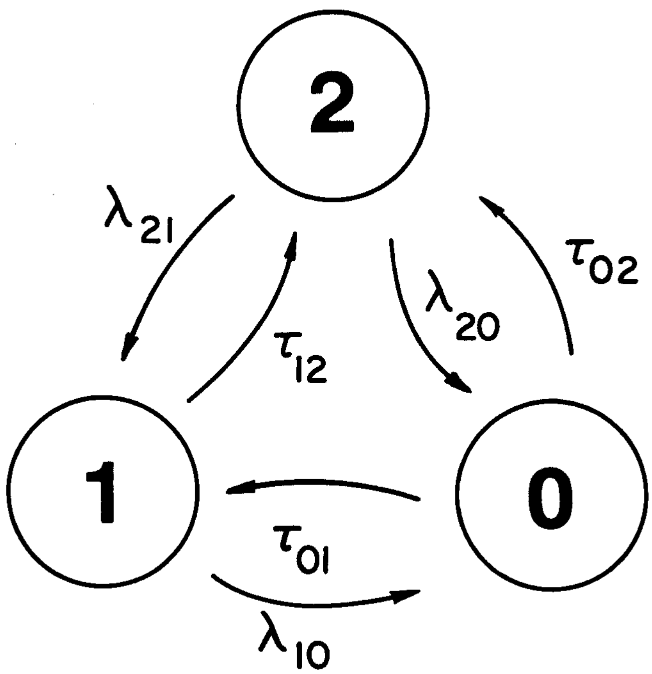

Figure 1: State transition diagram.

- A component in state 1 may fail into state 0 with failure rate λ_{10}.
- A component has only one failure mode.
- Three types of failures are considered: demand, dependent, independent failures. The demand failures may occur when a component, which is on-line but not in operation, is demanded to operate at the startup of the entire system. The dependent failures occur with a specific probability when particular components fail from state 2 to 0. The independent failures are failures of components during operation and the failures do not affect the states of other components.

2.3 Unscheduled Maintenance Model

- A component in state 0 can be repaired into state 1 or 2 within a specific time interval, τ_{01} or τ_{02}.
- A component in state 0 can be replaced by a new unit and goes to state 1 or 2 within a specific time interval, τ_{00}.
- A component in state 1 can be repaired into state 2 within a specific time interval, τ_{12}, by shutting it down, i.e. state 0, or keeping the state in 1.
- A component in state 1 can be replaced by a new unit and goes to state 2 within a specific time interval, τ_{11}. During the replacement the state of the component is 0.
- Whenever repair is required, the availability of repair facilities for the particular component is asked. If no facility is available, the repair is postponed until the facility becomes available.
- Before replacement is actually performed, the availability of spares is asked. If there is no spare available, the component is repaired instead of being replaced.
- The repair of a component can be deferred until the next scheduled maintenance of this component or the next complete shutdown of the entire system

- A subsystem represented as a gate in a systems tree can be replaced by a new subsystem with a specific time interval. Such a subsystem cannot contain other subsystems but only components.

- A repair of a particular component may affect the states of other components. But after the repair is completed, the states of components affected resume to state 2.

- The program uses repair times given by: $\tau_{new} = \Delta t * [\tau/\Delta t + 0.5]$, where τ is data, Δt is the timestep used in the simulation, and $[x]$ indicates truncation of number x.

2.4 Scheduled Maintenance Model

- Scheduled maintenance is performed for a component. During the maintenance, the state of the component can be one of three states.

- If a component is being repaired or replaced when a scheduled maintenance period occurs, the repair/replacement is continued until finished. Once it is done, the normal scheduled maintenance is performed. If the repair/replacement is not done within the scheduled maintenance period, it is continued even after the period.

- A deferred repair of a component is performed during a scheduled maintenance period or shutdown period of the entire system. If finished in this period, the normal scheduled maintenance is performed for the rest of the period. If the repair is not completed, it is performed during the next scheduled maintenance period. The state between the successive scheduled maintenance periods is kept at 0.

2.5 Economics Model

- The cost of a component is computed by adding the capital cost and the cost of equipment and personnel required by scheduled maintenance.

- The total cost of a system is the sum of the costs of components, the costs of unscheduled maintenance equipment and personnel, and the costs of spares used during an operation.

3 Input and Output Information

3.1 Description of Input Data

1. Job Title Card

The user enters the title of the job in an A72 format.

2. Namelist Input

(ICCOST=1,IDCONT=0, etc.. End with symbol $.)

variable name	type	description
ICCOST	I	0/1 no/yes economic data input and computation.
IDCONT	I	0/1 no/yes continued run.
IBIAS	I	0/1/2/3 method of variance reduction, 0= no method is applied, 1= method I, 2= method II, 3= method III.
	(NOTE)	No technique is available for the present version.
ITSRTE	I	0/1 no/yes compute mean-time-to-state. transitions
IPLTRE	I	0/1 no/yes plot the systems tree.
IPLTST	I	0/1 no/yes plot the time-dependent state of the top gate for the first history.
IPLTDF	I	0/1 no/yes plot the distribution functions of the top gate and create a file "ofdf".
IPLAVT	I	0/1 no/yes plot the time-dependent e-avail of the top gate and create a file "otev".
IDEMND	I	0/1 no/yes demand failures exist.

IRPFAC	I	0/1/2 maintenance facility option. 0= no, 1= limited availability of repair facility, 2= compute the maximum amount of repair facility.
ISPARE	I	0/1/2 spare option 0= no, 1= limited amount of spares, 2= compute the amount of spares required for an operation.
IMPRTC	I	0/1 no/yes compute importances.
ISNVTY	I	0/1 no/yes compute sensitivity coeffs.

The default values of the above twelve variables are 0.

NENTMX	I	maximum input numbers for non-virtual gates (default = 20).
NCOMP	I	Number of components.
NGATE	I	Number of gates.
LEVEL	I	Number of gate levels in the systems tree.
MANTMX	I	The maximum number of scheduled maintenance.
NEQUIP	I	Number of types of maintenance facilities.
NHIST	I	Total number of histories for a simulation.
NPRNT	I	Frequency of printouts of the state availabilities of the top gate. They are printed out every NPRNT histories.
TRENEW	R	Time interval for spare supply; spares are supplied
TSTEP	R	The timestep length (hours).
TOTAL	R	The total time length of a simulation for one history (hours).

3. Input of Remaining Data

CARD 1 (NGATE cards are always required).

variable name	length	format	description
NOGATE(1,n)	1	I3	ID number of gate n.
NAMEG(n)	1	A12	Name of gate n.

NOGATE(2,n)	1	I3	Level of gate n.
NOGATE(3,n)	1	I3	Number of inputs to gate n.
NOGATE(4,n)	1	I3	-1/1/2/3 user specified gate logic/ AND/SUM/MAX.
NOGATE(5,n)	1	I3	s_1 for SUM logic.
NOGATE(6,n)	1	I3	s_2 for SUM logic.
NOGATE(7,n)	1	I3	0/-1/-2 gate replacement option. 0= no replacement, 1= replaced if it is in state 0, 2= replaced if it is in state 1.
RETNTG(n)	1	I3	Number of spares for gate n.
RPLTMG(n)	1	F10.0	Time length required for replacement.
COEFG(1,n)	1	F5.0	Coeff. γ for effective availability.
COEFG(2,n)	1	F5.0	Coeff. β for effective availability.
COEFG(3,n)	1	F5.0	Coeff. α for effective availability.

(NOTE) The effective availability, A, is defined by

$$A = \alpha A_2 + \beta A_1 + \gamma A_0$$

where A_i (i=0,1,2) is the i-th state availability.
(default: $\alpha = 1.0$, $\beta = 0.5$, and $\gamma = 0.0$.)

CARD 2 (input if necessary).

LOGIC	3^m	4X,36I2	Logic for gate with m inputs. Input if NOGATE(4,n)=-1.

(NOTE) The order of input should be
$(...((LOGIC(i_1, i_2, ..., i_m), i_1 = 1,3), i_2 = 1,3), ...), i_m = 1,3)$.

CARD 3&4 (NGATE cards are always required).
CARD 3.

-	1	3X	Gate ID.
NENTG(n)	1	I3	Number of gate inputs to gate n.
NENTC(n)	1	I3	Number of component inputs to gate n.

CARD 4.

INSYS(i,n)	NENTG(n) +NENTC(n)	3X,20I3	ID number of input gates and components.

CARD 5. (NCOMP cards are always required).

KDCOMP(1,i)	1	I3	ID number of component i.
NAMEC(i)	1	A12	Component name.
KDCOMP(2,i)	1	I3	Number of identical units.
	(NOTE) on-line redundancy		
KDCOMP(3,i)	1	I3	1/2/3 logic of virtual gate. AND/SUM/MAX.
KDCOMP(4,i)	1	I3	s_1 for SUM logic.
KDCOMP(5,i)	1	I3	s_2 for SUM logic.
KDCOMP(6,i)	1	I3	Repair/replace type for state 0. -2= replace to get state 2, -1= replace to get state 1, 1= repair to get state 1, 2= repair to get state 2, 3=defer repair until sch. maint., 4= defer repair until system down.
KDCOMP(7,i)	1	I3	Repair/replace type for state 1. -1= replace, 0= fail, 1= repair by shutting it down, 2= repair by keeping the state, 3= defer repair until sch. maint., 4= defer repair until system down.
KDCOMP(8,i)	1	I3	Failure mode type. -1= demand failure, 0= independent failure, n= dependent failure; it affects n components (only for 2 to 0 transition).
KDCOMP(9,i)	1	I3	-1= entire system must

			be shut down.
			n= number of components affected.
KDCOMP(10,i)	1	I3	Number of scheduled maintenance.
REDNTC(i)	1	I3	Number of spares.
COEFC(1,i)	1	F5.0	Coeff. γ for effective availability.
COEFC(2,i)	1	F5.0	Coeff. β for effective availability.
COEFC(3,i)	1	F5.0	Coeff. α for effective availability.
ENHANCE(i)	1	F10.0	Importance factor.
			Required if IBIAS=0.

CARD 6,7,& 8 (NCOMP cards are always required).

CARD 6.

-	1	4X	Component ID.
TRMX(1,1,i)	1	F12.0	τ_{00}.
TRMX(2,1,i)	1	F12.0	λ_{10}
TRMX(3,1,i)	1	F12.0	λ_{20}

CARD 7.

-	1	4X	blanks.
TRMX(1,2,i)	1	F12.0	τ_{01}
TRMX(2,2,i)	1	F12.0	τ_{11}
TRMX(3,2,i)	1	F12.0	λ_{21}

CARD 8.

-	1	4X	blanks.
TRMX(1,3,i)	1	F12.0	τ_{02}
TRMX(2,3,i)	1	F12.0	τ_{12}
TRMX(3,3,i)	1	F12.0	not used.
DATABS(i)	1	A12	Name of data base.

CARD 9 (If necessary).

-	1	4X	Component ID.
IDCOMC(j,i)	KDCOMP(8,i)	10I5	Component IDs that may fail after failure of component i.

CARD 10 (If necessary).

CONDPR(j,i)	KDCOMP(8,i)	10F8.0	Conditional probabilities

for dependent failure.

CARD 11 (If necessary).

-	1	4X	Component ID.
IDRPLC(j,i)	KDCOMP(9,i)	10I5	Component IDs affected by repair of component i.

CARD 12 (If necessary).

NSTRPL(j,i)	KDCOMP(9,i)	10I5	State of component j during repair of component i.

CARD 13&14 (If necessary).

CARD 13.

-	1	4X	Component ID.
MAINTYPE(j,i)	KDCOMP(10,i)	36I2	States of component i during j-th scheduled maintenance.

CARD 14.

TMLINE(1,j,i)			Start and end times of
TMLINE(2,j,i)	KDCOMP(10,i)	10F8.0	j-th sch. maint. (hours)

CARD 15 (NEQUIP cards are required if IRPFAC≥1).

-	1	4X	Facility ID.
ENAME(n)	1	A12	Name of maint. facility.
EQUIP(n)	1	I5	Number of maint. facility n.

CARD 16 (NCOMP cards are required if IRPFAC≥1).

-	1	4x	Component ID.
NEEDS(k,i)	NEQUIP	20I3	Amount of repair facility k needed by component i.

CARD 17 (NCOMP cards are required if ICCOST=1).

-	1	4X	Component ID.
CAPCST(i)	1	F10.0	Capital cost of component i.
SCHMTCT(i)	1	F10.0	Scheduled maint. cost.

CARD 18 (NEQUIP cards are required if ICCOST=1 and IRPFAC≥1.)

-	1	4X	Maintenance facility ID.
CAPCST(i)	1	F10.0	Cost of maint. equipment i.

3.2 Description of Output

1. **Echo of input data**

2. **Results of simulation**

   - The state availabilities and uncertainties of the top gate printed out every NPRNT histories.
   - The state availabilities and uncertainties of gates and components.
   - The effective availability of the entire system.
   - If ITSRTE=1, mean-time-to-state transitions.
   - The ranking of gates and components according to the magnitudes of effective availabilities. The 95% confidence intervals are also printed.
   - If IRPFAC=2, the maximum number of maintenance facilities at a certain time and the 95% confidence intervals.
   - If ISPARE=2, the number of spares required by an operation and 95% confidence intervals.
   - If ICCOST=1, the capital cost per unit, the cost of scheduled maintenance, and the total cost of components.
   - If ICCOST=1 and IRPFAC$\geq$1, the cost of maintenance facilities.
   - If ICCOST=1, the net cost of the entire system in dollars.
   - If IMPRTC=1, components' importances are computed and printed in order of magnitude.
   - If ISNVTY=1, sensitivity coefficients with respect to state transition parameters are computed and printed.
   - According to values of IPLTRE, IPLTST, IPLTDF, and IPLAVT, corresponding figures are plotted in graphics files.

3. Run information

- The amount of computer memory required by the run.
- Computing time in seconds: CPU, I/O, and SYSTEM.

4 Program Information

4.1 Program Structure

1. Program flow

The flowchart and program structure of the PROPA program are given in Figs. 2 and 3, respectively.

2. Role and Function of Subprograms

GATEDAT reads data on gates from the input file and writes the data on the output file.

TREE reads data on interconnections among components and gates required to construct a systems tree. The gate IDs are rearranged in such an order that the top gate comes first, the gates in the second level of the tree come next and so on. The ID of the top gate is identified.

PLTTRE draws the systems tree. The plot is made by using the DISSPLA library.

PLTGAT draws triangles that represent gates.

RELDAT reads and writes data on components. It creates the INTUT array, which indicates the first position of quantities of one of identical units in variable arrays. It computes the state transition probabilities by using the input data on failure. See Ref.[1] about equations used. It reads and writes data on dependent failures. When a variance reduction technique (Method I) is used, this computes weighted transition probabilities.

MAINTDAT reads and writes data on scheduled maintenance for components.

FACTDAT reads and writes data on maintenance facility.

COSTDAT reads the economics data of the system.

CALC performs a Monte Carlo simulation. At the end of a history, this computes state availabilities, uncertainties, and covariances. After all histories are completed, it writes these quantities in binary file OFILE. The flowchart of this program is given in Fig. 4.

HIST computes the states of components (DO loop 1000) and gates (DO loop 2000) at the end of the current timestep. The flowcharts are given in Figs. 5, 6, and 7.

STORE updates the amount of spares every TRENEW hours. This is called if ISPARE=1.

FAILURE checks if a component fails during the current timestep.

UNSCHMAN checks if a component is repaired/replaced during the current timestep. The actions taken in subprograms FAILURE and UNSCHMAN can be seen in Fig. 8.

SCHMANT sets the states of components to specific states when a scheduled maintenance period comes. This also checks if deferred repairs are performed and if a component is already in a repair/replacement stage. The flowchart of this subprogram is given in Fig. 9.

MANTFAC checks if maintenance facilities for a component are available. Those facilities are returned at the end of a repair so that they can be used by other repairs. This routine is called if IRPFAC$\geq$1. The flowchart is given in Fig. 10.

SPARES checks if a spare is available for a replacement. This routine is called if ISPARE $\geq$1. The flowchart is given in Fig. 11.

NGATE33 computes the output state of a gate from input states by using the logic specified by the user. See Appendix A about gate logic.

DEPFLR checks if dependent failures occur.

DEPRPL checks if there are effects on other components during the repair of particular components.

PLTSTATE plots the time-dependent state of the top gate for the first history by using the DISSPLA library.

EDIT writes final results. It also computes the effective availabilities of gates and components and confidence intervals. It makes the ranking of gates and components.

SORT does sorting for the ranking of gate and component availabilities.

CALCOST writes economics data and computes and writes costs of components, maintenance facilities, and spares. Finally the cost of the entire system is computed and written. This routine is called if ICOST=1.

ADJST computes adjusted durations for which components and gates are in particular states. This is used if IBIAS=1.

RANDNO generates random numbers by using the recursive congruential method[3].

PFCT computes component importances (See Appendix B).

SENSIT computes sensitivity coefficients of components with respect to state transition parameters (See Appendix B).

DFGRAPH plots the distribution functions of top gate availability.

AVGRAPH plots the time-dependent e-avail and its 95% confidence intervals.

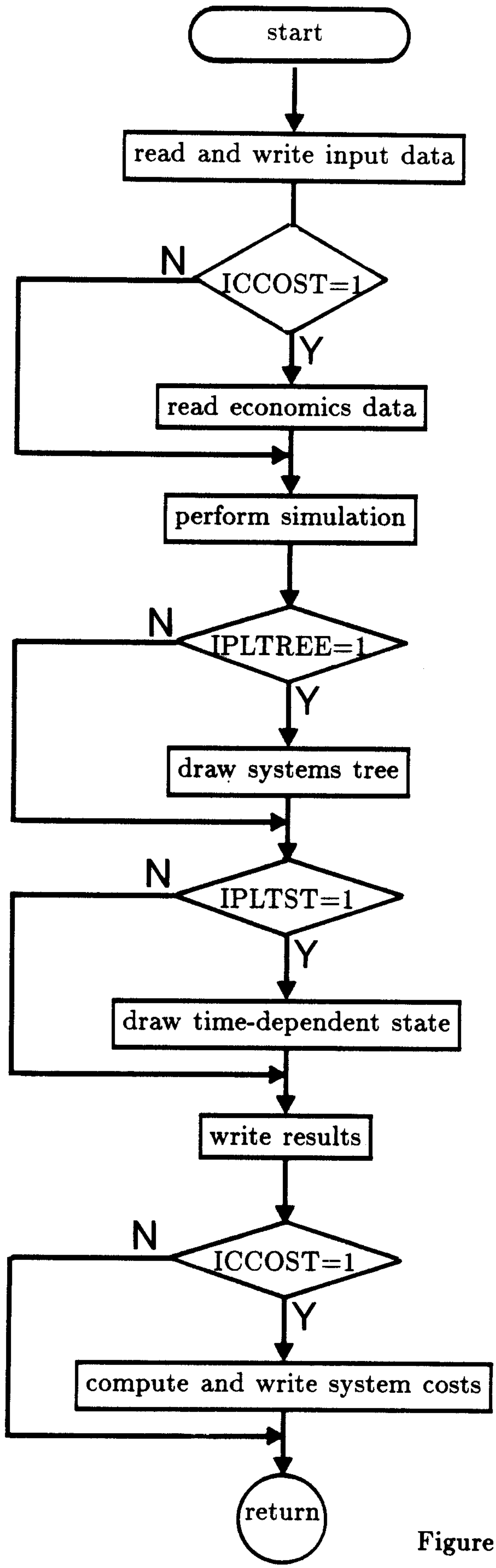

Figure 2: Flowchart of PROPA.

Figure 3: Program structure of PROPA.

A. MAIN

B. Input Module
  1. GATEDAT
  2. TREE
  3. RELDAT
  4. MAINTDAT
  5. FACTDAT
  6. COSTDAT

C. Simulation Module
  1. CALC
     i. HIST
        a. SCHMANT
        b. UNSCHMAN
        c. FAILURE
        d. NGATE33
        e. ADJST
        f. SPARES
        g. MANTFAC
        h. DEPFLR
        i. DEPRPL
        j. RANDNO
        k. PFCT
     ii. STORE

D. Edit Module
  1. PLTTRE
     i. PLTGAT
  2. PLTSTATE
  3. DFGRAPH
  4. AVGRAPH
  5. SENSIT
  6. EDIT
     i. SORT
  7. CALCOST

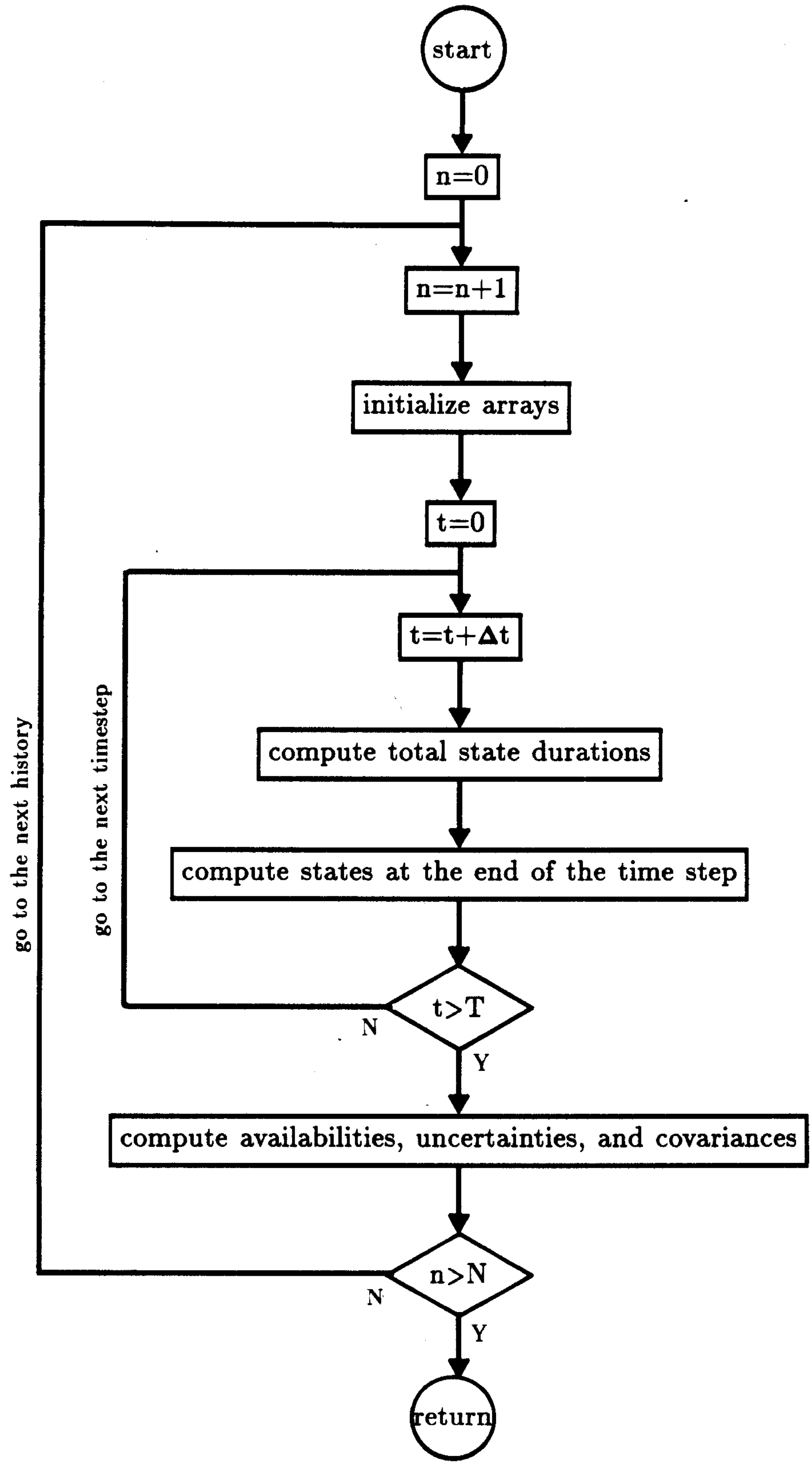

Figure 4: Flowchart of subprogram CALC.

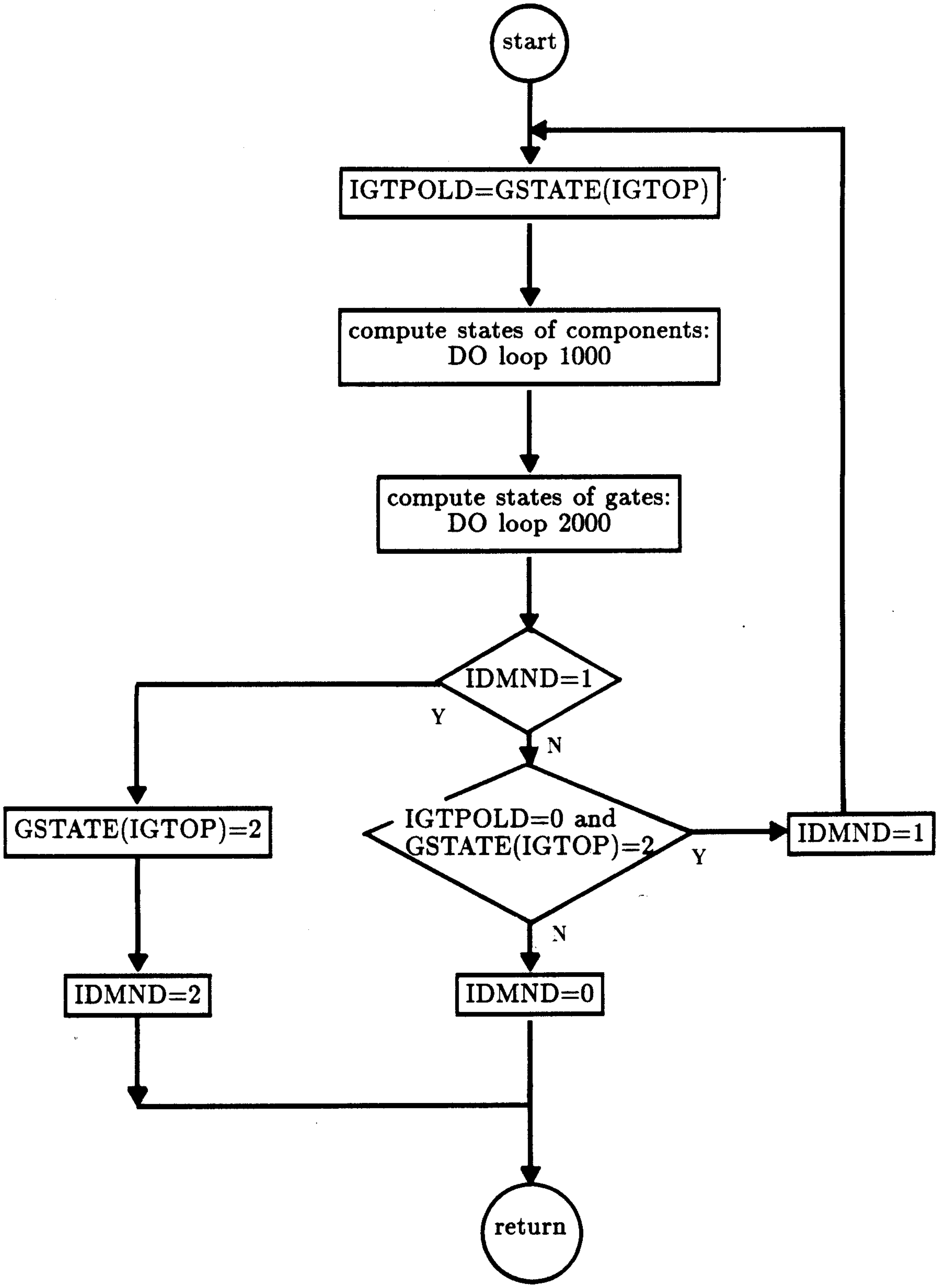

Figure 5: Flowchart of subprogram HIST.

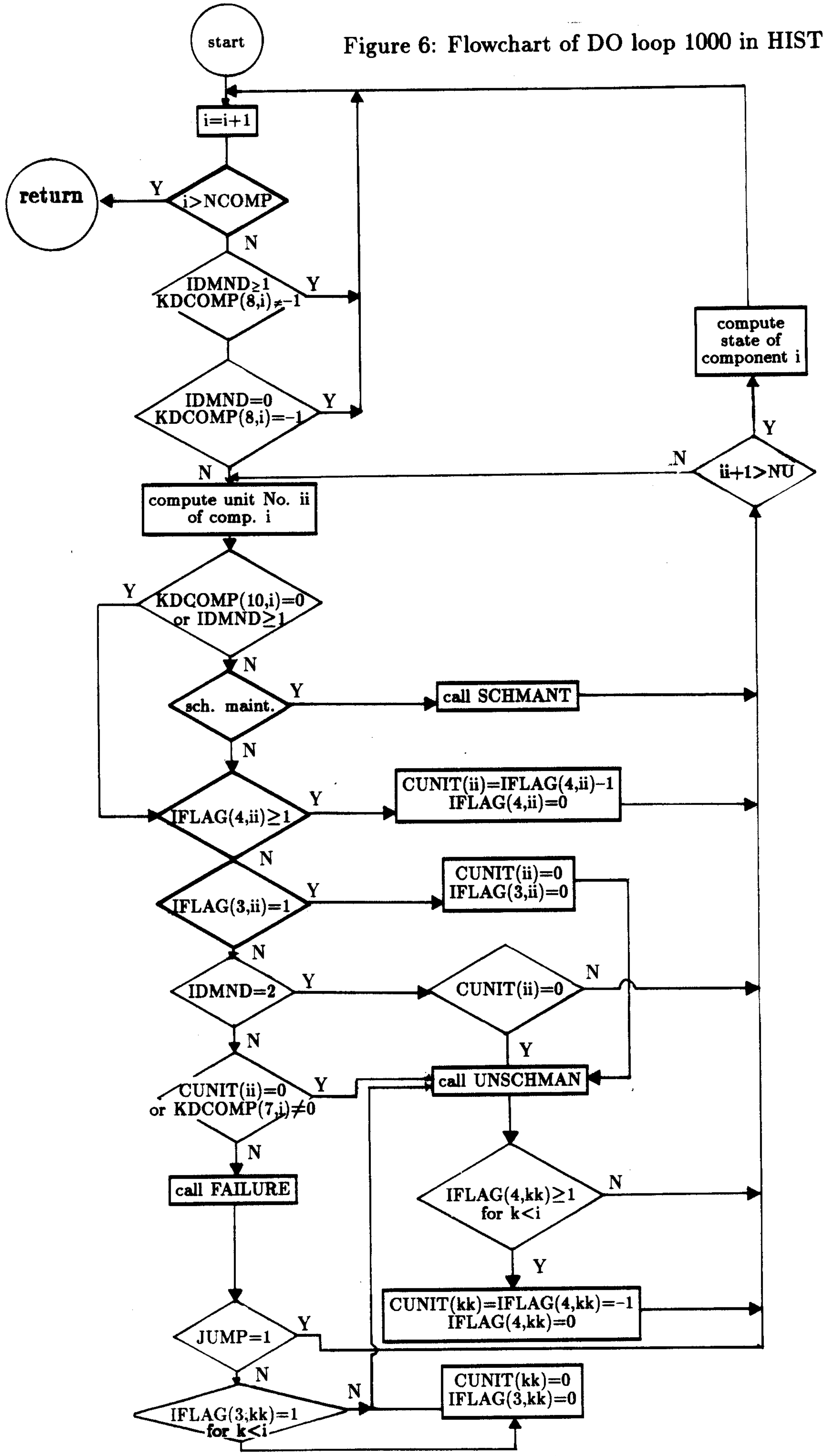

Figure 6: Flowchart of DO loop 1000 in HIST.

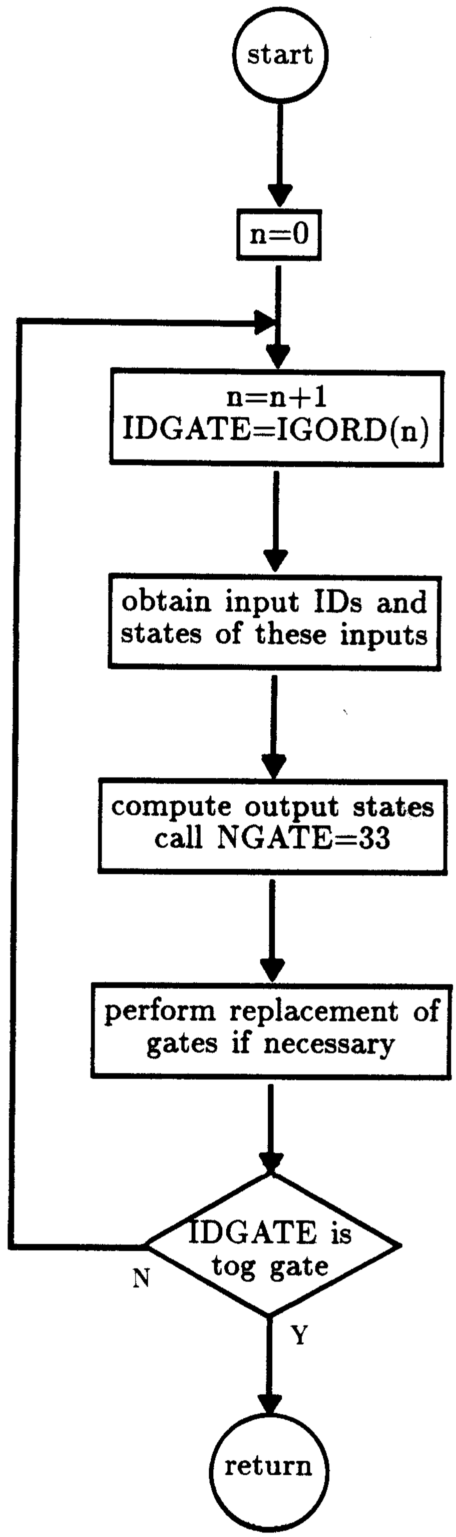

Figure 7: Flowchart of DO loop 2000 in HIST.

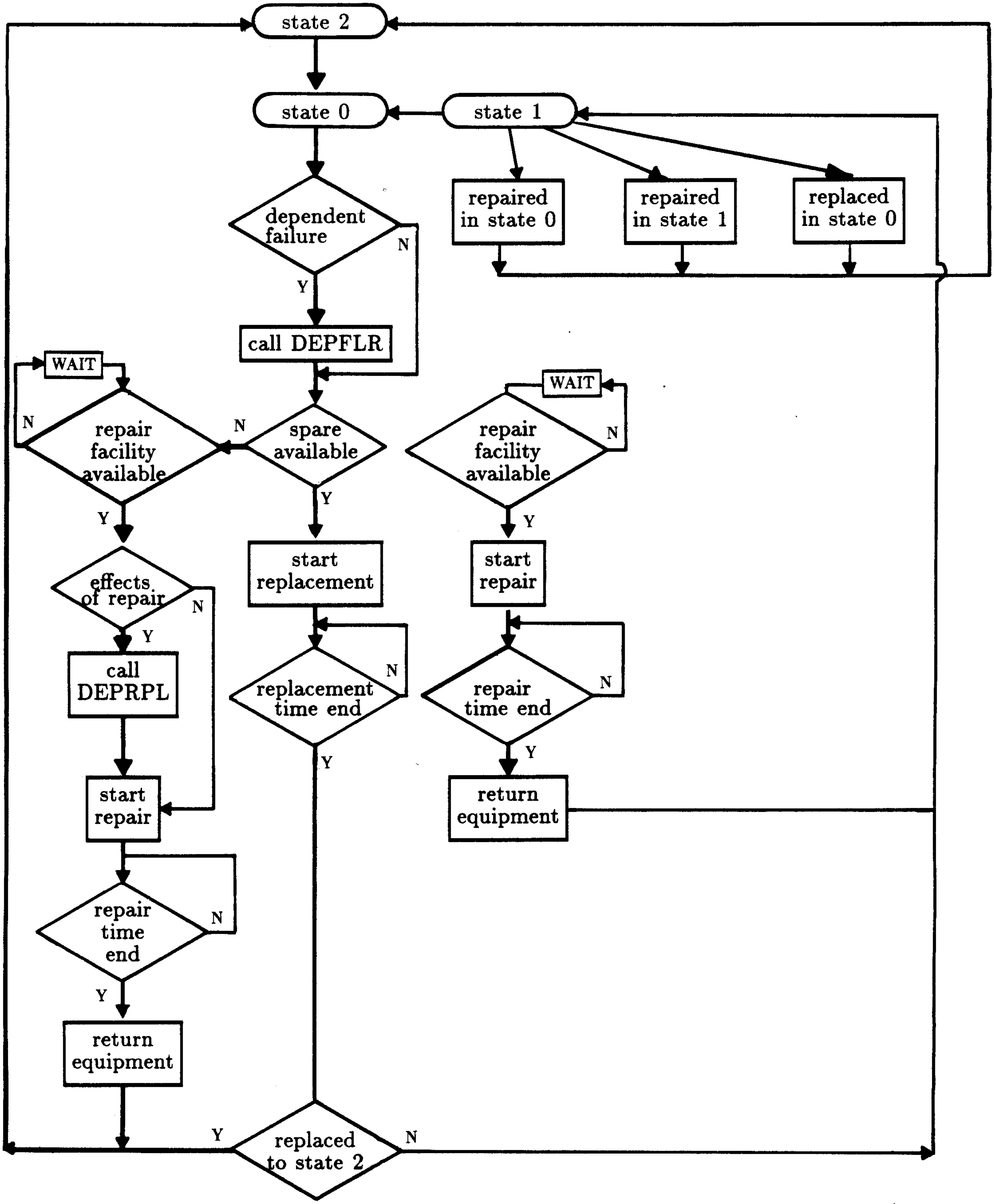

Figure 8: Failure and repair sequences.

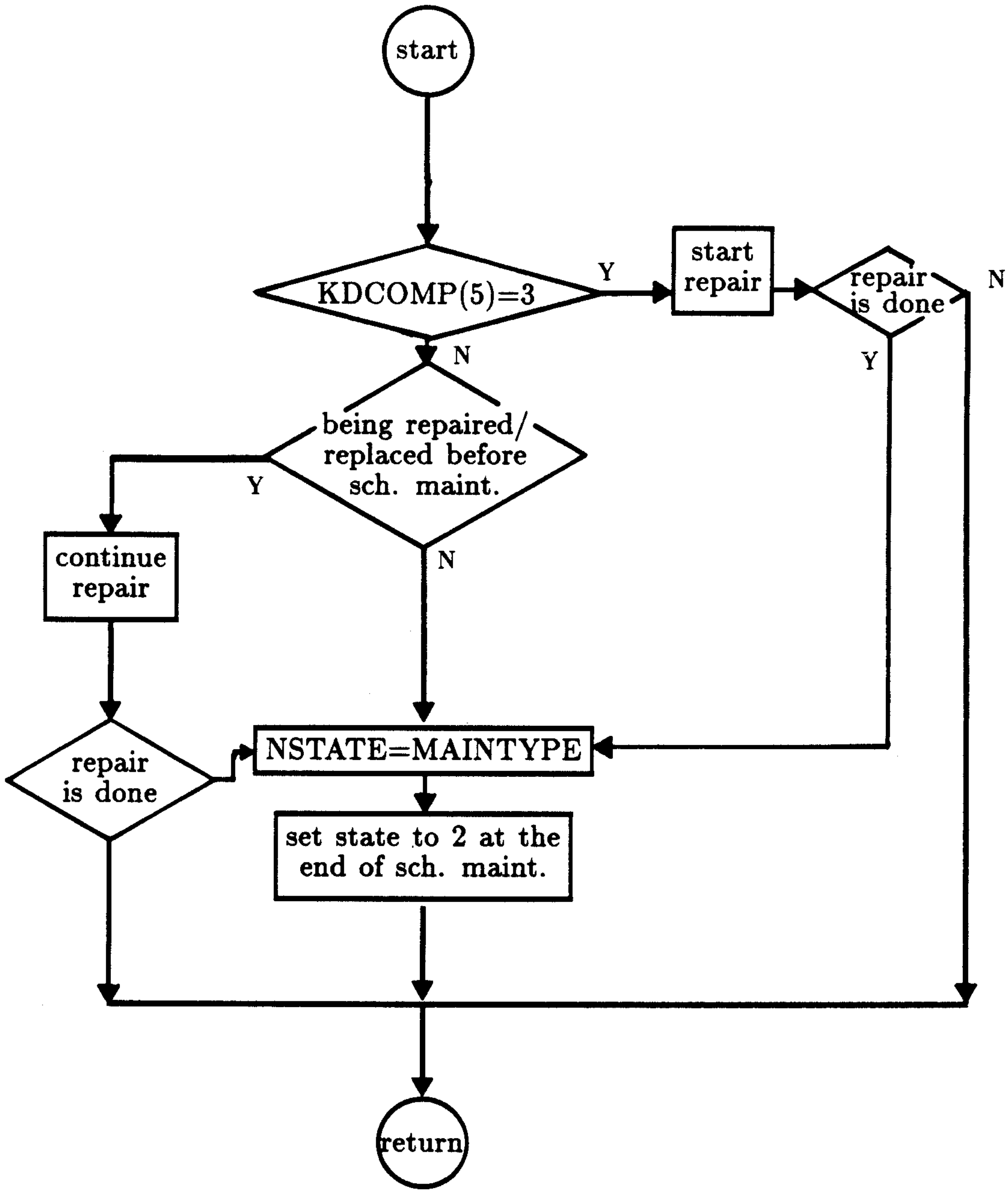

Figure 9: Flowchart of subprogram SCHMANT.

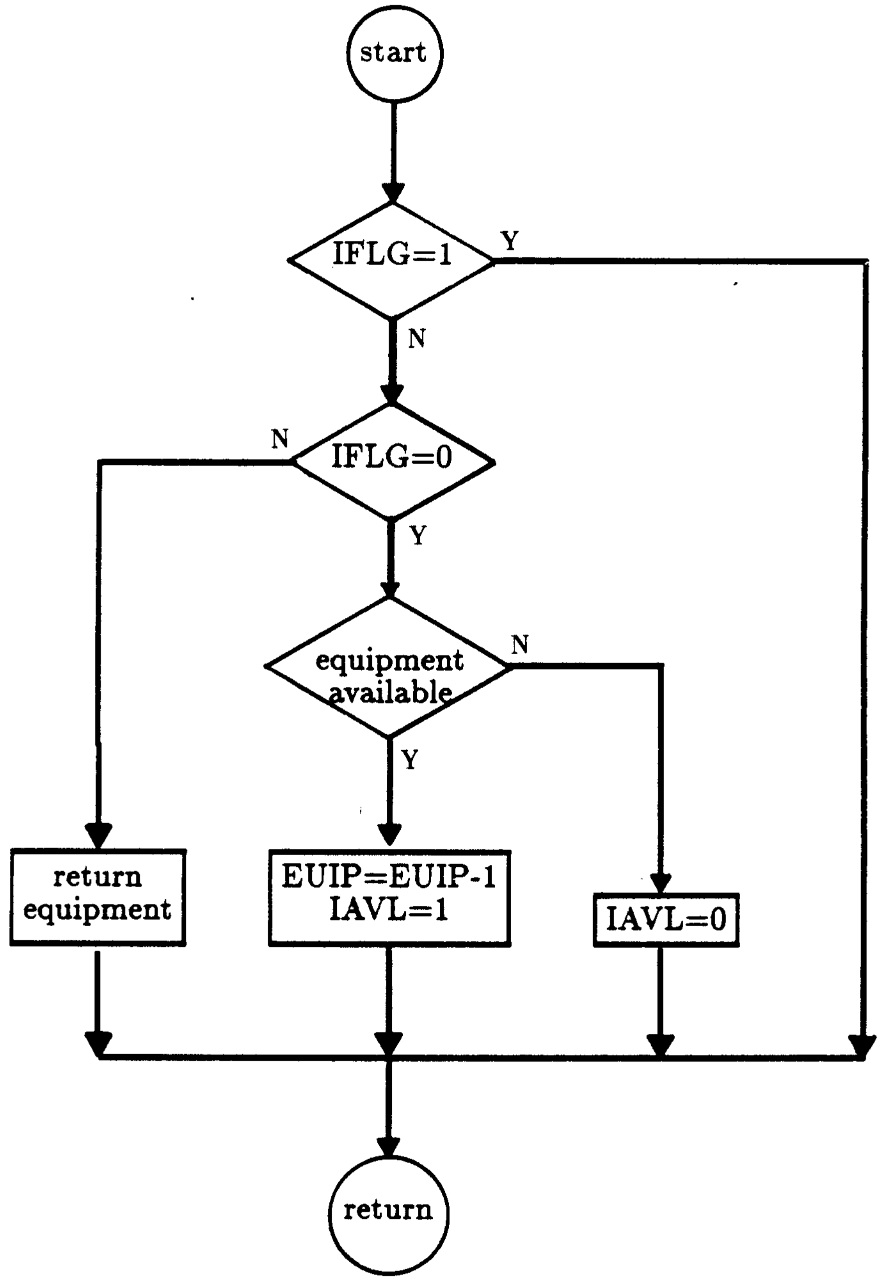

Figure 10: Flowchart of subprogram MANTFAC.

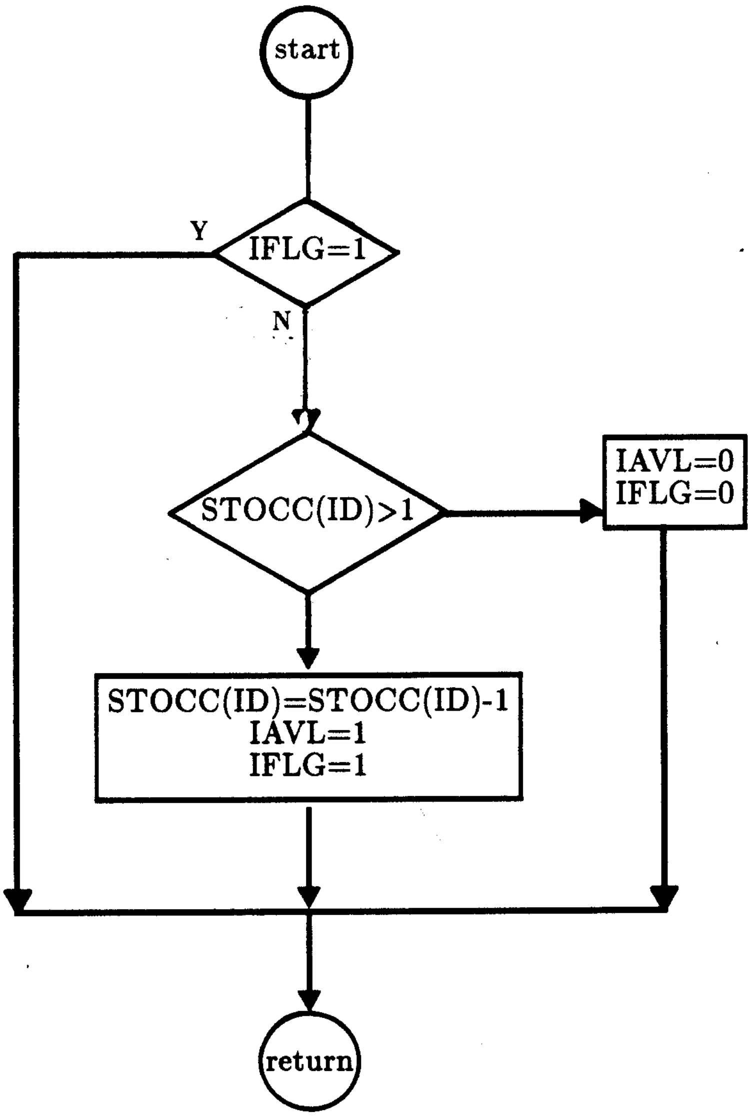

Figure 11: Flowchart of subprogram SPARES.

4.2 External Files

File name	Description
I	Input file.
O	Output file.
OFILE	Binary output file.
BFILE	Binary input file. OFILE must be renamed BFILE for a continued run.
OFDF	output file containing the distribution functions of top-gate availability.
OTEV	output file containing the time-dependent e-avail of top gate.

4.3 Hardware Requirements

The current PROPA program is being executed on a CRAY-1 computer. The size of the executable program is about 1.2 MBytes with array A(90000) that indicates the maximum allowable memory size.

4.4 Software Requirements

The program was written by using a modified FORTRAN77 for the CFT compiler on CRAYs[4]. The DISSPLA library[5] is required for graphics options. It should be noted that the program assumes that values of all variable arrays are set to zero by a compiler.

4.5 Availability of PROPA at NMFECC

Read an executable program by

```
FILEM  RDS  1751  .PROPA16  PRXXOOOY
```

Here XX, OOO, and Y represent the date, month, and year when the program is created; for example, PR13AUG5 was created on August 13, 1985.

In the same directory, the file PROPA16 is the source listing.

To run it, simply type in

```
PRXXOOOY  I=" inputfile name",O=" outputfile name"
```

The graphs are contained in a file with name "faxxxxxx". To plot this graphics file on a Versatec, type in

```
NETPLOT  faxxxxxx
```

Acknowledgement

Support for this work has been provided by the U.S. Department of Energy.

References

[1] Y. Watanabe,"A Monte Carlo Simulation Method for Systems with a Degraded State," University of Wisconsin Fusion Technology Institute Report UWFDM-644 (November 1986)

[2] Y. Watanabe,"Availability Analysis of Fusion Reactor Plants with Degraded States," University of Wisconsin Fusion Technology Institute Report UWFDM-646 (August 1985)

[3] D. E. Knuth,"The Art of Computer Programming, Vol. 2," Addison-Wesley, Reading, MA (1981)

[4] Cray Research, Inc., "CFT, the CRAY-1 FORTRAN compiler", available at the National Magnetic Fusion Computer Center, Livermore, CA (1984)

[5] ISSCO,"DISSPLA User's Manual, Version 9.2," Integrated Software Systems Corp., San Diego, CA (1984)

Appendix A: Definition of Gate Logic

We consider gates with I inputs. The state of the output and the state of the i-th input are denoted by y and x_i, respectively.

AND gate

$$y = Min\{x_i\}. \quad (1)$$

SUM gate

$$y = \begin{cases} 0 & if \quad 0 \leq X \leq s_1 \\ 1 & if \quad s_1 < X \leq s_2 \\ 2 & if \quad s_2 < X \leq 2I \end{cases} \quad (2)$$

where $X = \sum_{i=1}^{I} x_i$.

MAX gate

$$y = Max\{x_i\}. \quad (3)$$

Appendix B: Sensitivity Analysis Method

The effective availability (e-avail) of entire system, A, is given by

$$A = \phi(a_1, a_2, \cdots, a_n) \tag{1}$$

where a_k is the e-avail of component k. Furthermore, a_k is a function of state transition parameters of component k:

$$a_k = f(\theta_{ki}) \tag{2}$$

where $\theta_{ki} = \lambda_{20}, \lambda_{21}, \lambda_{10}, \tau_{01}, \tau_{12}, or\ \tau_{02}$.

It is necessary to know the sensitivity of A to small variations in θ_{ki}; in other words, we need to compute

$$\frac{\partial A}{\partial \theta_{ki}} \tag{3}$$

This can be done by computing

$$\frac{\partial A}{\partial a_k} \tag{4}$$

and

$$\frac{\partial a_k}{\partial \theta_{ki}} \tag{5}$$

In the program the derivative (4) for component k, which is called importance, is computed by a Monte Carlo simulation using the following estimator S_k:

$$S_k^{(n)} = \frac{1}{2J} \sum_{j=1}^{J} \{\psi(x_1, \cdots, x_{k-1}, 2, x_{k+1}, \cdots, x_n) - y_j\} \tag{6}$$

$$S_k = \frac{1}{N} \sum_{n=1}^{N} S_k^{(n)} \tag{7}$$

where x_i is the state of component i (i.e. $x_i = 0$, 1, or 2), J is the number of time steps for a simulation, and N is the number of histories. y_j is the state of entire system and given by

$$y_j = \psi(x_1, x_2, \cdots, x_n) \tag{8}$$

The second derivative (5) is not computed by simulation; an analytical approach is taken. In steady-state, state availabilities of component i, P_i, are the solution of the following equations:

$$(\lambda_{20} + \lambda_{21})P_2 = \mu_{02}P_0 + \mu_{12}P_1 \tag{9}$$

$$(\lambda_{10} + \mu_{12})P_1 = \lambda_{21}P_2 + \mu_{01}P_0 \tag{10}$$

$$P_0 + P_1 + P_2 = 1 \tag{11}$$

The e-avail is given by

$$a_k = \alpha P_2 + \beta P_1 \tag{12}$$

where γ is set to 0. Thus the sensitivity coefficient, s_{ki}, which is the sensitivity of e-avail of component k to i-th parameter, can be given by

$$s_{ki} = \alpha \frac{\partial P_2}{\partial \theta_{ki}} \frac{\theta_{ki}}{P_2} + \beta \frac{\partial P_1}{\partial \theta_{ki}} \frac{\theta_{ki}}{P_1} \tag{13}$$

In practice the analytical formulas of Eq. (13) are obtained by using a symbolic manipulation program REDUCE and implemented in subprogram SENSIT.

Appendix C: Mean-Time-to-State Transitions

It is useful to calculate mean-time-to-failure (MTTF) and mean- time-to-repair (MTTR) of the entire system because these parameters can be used to estimate the availability of other systems that include the system as a component. Such an approach certainly reduces simulation time.

In the three-state model, we compute mean-time-to-state transitions (MTTST) for six types of state transitions, i.e. state 2 to 1, 2 to 0, 1 to 0, 0 to 1, 0 to 2, and 1 to 2. The MTTSTs for state 2 to 0 and 0 to 2 are equivalent to MTTF and MTTR, respectively.

The following estimator for the MTTST for state i to j, τ_{ij}, is used:

$$\tau_{ij} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{K_{nij}} \sum_{k=1}^{K_{nij}} T_{ij}^{(k)} \tag{1}$$

where N is the number of histories, K_{nij} is the number of transitions from state i to j during the n-th history, and $T_{ij}^{(k)}$ is the time length until the k-th transition from state i to j occurs for the first time since the state changed to state i.

In the program, τ_{ij} and the standard deviations are computed for the top gate, i.e. entire system.

Appendix D: Sample Input

A sample input file is reproduced here. The system consists of two gates and four components.

```
inp14 : debugging input for propa16
ncomp=4,ngate=2,level=2,mantmx=2,nequip=2,
nhist=50,nprnt=10,tstep=48.0,total=8760.0,
ipltre=0,ipltst=0,idemnd=1,irpfac=2,ispare=2
iccost=1,imprtc=1,isnvty=1
$
  1 system        1  3  1  0  0  0  0
  2 power         2  2  1  0  0  0  0
  1  1  2
     2  3  4
  2  0  2
     1  2
  1 resistance    1  0  0  0  2  0  1  1  2  0
  2 battery       1  0  0  0  2  0  0  0  2  0
  3 switch        1  0  0  0 -2  0  0  0  2  0
  4 wires         1  0  0  0  2  0 -1  0  0  0
   1           0.0       2.0e-3       1.0e-3 arbitary
               0.0          0.0       1.0e-3 arbitary
              96.0          0.0          0.0 arbitary
   2           0.0       1.0e-4       1.0e-4 arbitary
               0.0          0.0       1.0e-4 arbitary
             192.0          0.0          0.0 arbitary
   3           0.0      1.0e-99       1.0e-3 arbitary
               0.0          0.0      1.0e-99 arbitary
              48.0          0.0          0.0 arbitary
   4           0.0      1.0e-99       1.0e-2 arbitary
               0.0          0.0      1.0e-99 arbitary
              48.0          0.0          0.0 arbitary
   1     2
```

```
   0.9
1    3
0
1 0 0
300.0   400.0   720.0   768.0
2 0 0
300.0   400.0   720.0   768.0
3 0 0
300.0   400.0   720.0   768.0
1 equipment 1    0
2 equipment 2    0
1  1  3
2  4  1
3  2  1
4  0  0
1       10.0       5.0
2        5.0       1.0
3       55.2      12.0
1       10.0
2       25.0
```

Appendix E: Sample Output

The output file for the input given in Appendix D is reproduced here. Plots of the systems tree, the time-dependent top gate state, the distribution function of top gate e-avail, and the time-dependent e-avail of top gate are also given in Figs. 12 to 15, respectively.

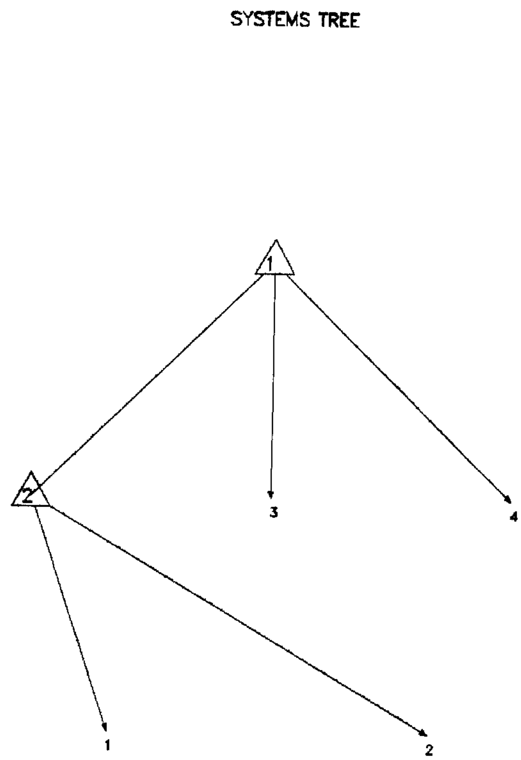

Figure 12: Schematic of a systems tree.

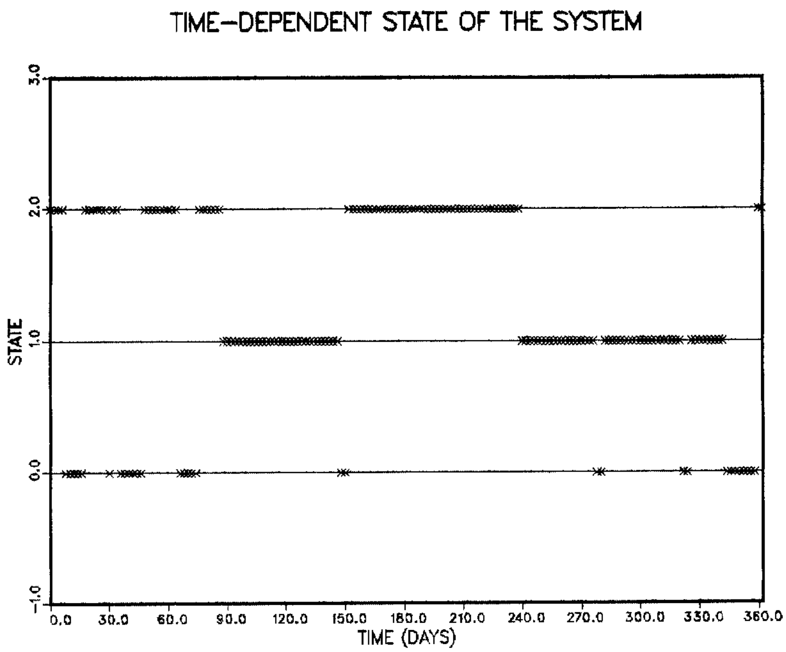

Figure 13: Time-dependent state of the top gate.

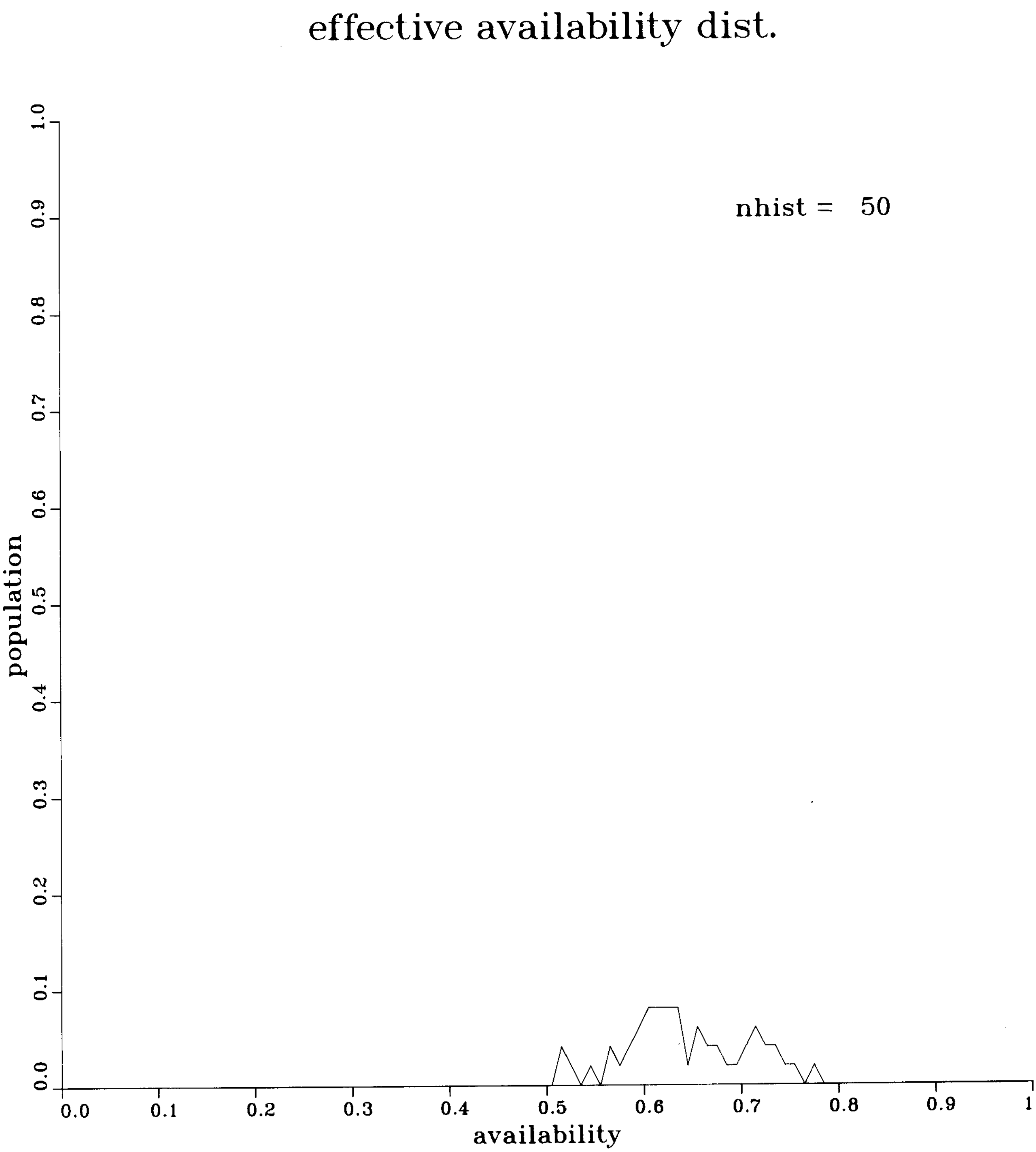

Figure 14: Distribution function of system e-avail.

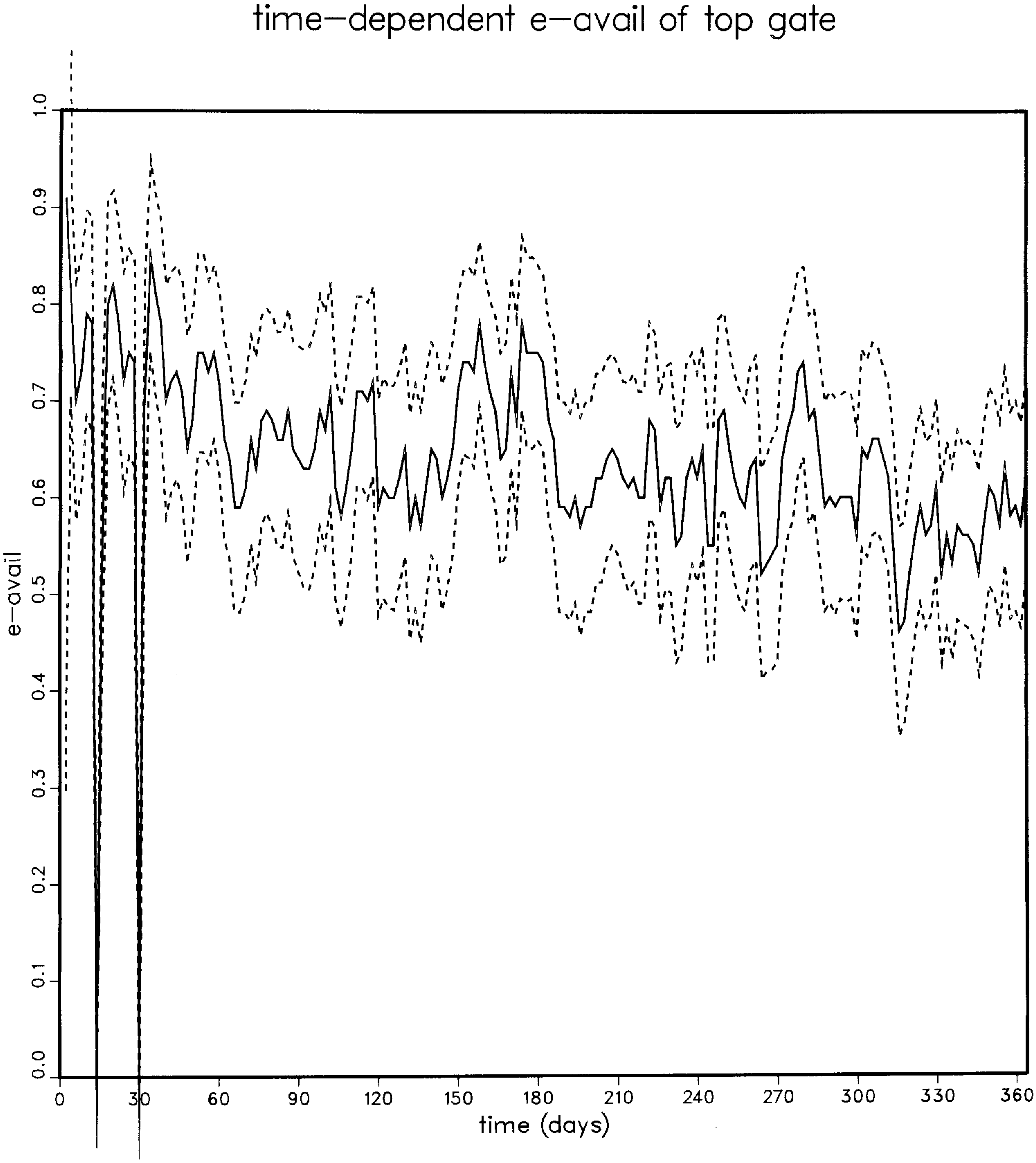

Figure 15: Time-dependent e-avail of the system.

```
******************************************************************************************
***   program propal.6   (12/09/86) *
                                                                        -10:09:19-12/09/86-d

 -  inp14 : debugging input for propa16

  (ncomp/ngate/level/nhist)=          4/          2/          2/         50/

 - time step for simulation =     48.000 hours
 - total time interval for simulation =   8760.000 hours


 - no variance reduction is utilized


 - gate information -

  ( gate logic type -1/1/2/3 = user sepcified/AND/SUM/MAX )

   ID  name          level  input no.  type  nsumg  nsumg  replace opt.  number      time for          coefg
                                             for 0  for 1  for 0/1       of spares   replace. hrs    0    1    2
    1  system          1        3        1     0      0       0            0            0.00       0.00 0.50 1.00
    2  power           2        2        1     0      0       0            0            0.00       0.00 0.50 1.00

 - systems tree data -

  gate ID     input gate ID/ input comp. ID
        1          2
                   3    4
        2
                   1    2

 - component data -

  ( virtual gate logic type 0/1/2/3=non/AND/SUM/MAX )

  ID    name      no. of    logic      nsum  for    repair    failure   effects    no. of    no. of         coefc          import.
                  units     type        0     1      0   1    mode      of repair  sch. man.   spares     0    1    2       factor
   1 resistance     1         0         0     0      2   0       1         1          2          0      0.00 0.50 1.00    0.00
   2 battery        1         0         0     0      2   0       0         0          2          0      0.00 0.50 1.00    0.00
   3 switch         5         2         1     5     -2   0       0         0          2          0      0.00 0.50 1.00    0.00
   4 wires          1         0         0     0      2   0      -1         0          0          0      0.00 0.50 1.00    0.00

 - 3 states transition matrices (input) -

 ID     (0,0)        (1,0)        (2,0)         (0,1)         (1,1)        (2,1)         (0,2)        (1,2)        (2,2)        data base
  1     0.0000   0.2000e-02   0.1000e-02       0.0000        0.0000   0.1000e-02      96.0000       0.0000   0.             arbitary
  2     0.0000   0.1000e-03   0.1000e-03       0.0000        0.0000   0.1000e-03     192.0000       0.0000   0.             arbitary
  3     0.0000   0.1000e-98   0.1000e-02       0.0000        0.0000   0.1000e-98      48.0000       0.0000   0.             arbitary
  4     0.0000   0.1000e-98   0.1000e-01       0.0000        0.0000   0.1000e-98      48.0000       0.0000   0.             arbitary
```

```
- input data on common mode failure -
 comp. ID   comp. ID/conditional prob.
         1         2
          0.9000

- input data on effects on other comps. during repair  -
comp. ID   comp. ID/state of the   component
        1     3
              0

- scheduled maintenace data -
    (state/start time/end time)

n\id          1                 2                 3                 4
 1  0    300.0    400.0  0    300.0    400.0  0    300.0    400.0  0   8808.0    8856.0
 2  0    720.0    768.0  0    720.0    768.0  0    720.0    768.0  0      0.0       0.0

- repair facility data -
 id  equipment name   no. of units
  1      equipment 1          0
  2      equipment 2          0

- equipments needed by components -
   comp\equip     1    2
          1       1    3
          2       4    1
          3       2    1
          4       0    0

*** memory required by this run ***
     (last/maxmem) =       2185/     90000


*** simulation starts ***

    history no.  availability and uncertainty of top event
                 state 0                     state 1                     state 2
          10     0.2470e+00 (0.2129e-01)  0.2689e+00 (0.2883e-01)  0.4842e+00 (0.3217e-01)
          20     0.2175e+00 (0.1564e-01)  0.3038e+00 (0.2274e-01)  0.4787e+00 (0.2357e-01)
          30     0.2140e+00 (0.1266e-01)  0.3102e+00 (0.1765e-01)  0.4758e+00 (0.1716e-01)
          40     0.2098e+00 (0.9926e-02)  0.3070e+00 (0.1603e-01)  0.4832e+00 (0.1512e-01)
```

50   0.2038e+00 (0.8411e-02)   0.3097e+00 (0.1588e-01)   0.4864e+00 (0.1492e-01)

*** final results **

(state 0 = failure/state 1 = degraded/state 2 = normal)

- gate ID

gate ID	name	availability state 0		state 1		state 2	
1	system	0.2038e+00	(0.8411e-02)	0.3097e+00	(0.1588e-01)	0.4864e+00	(0.1492e-01)
2	power	0.1717e+00	(0.7219e-02)	0.3097e+00	(0.1588e-01)	0.5186e+00	(0.1487e-01)

- component ID

component ID	name	availability state 0		state 1		state 2	
1	resistance	0.1164e+00	(0.4372e-02)	0.2701e+00	(0.1351e-01)	0.6136e+00	(0.1316e-01)
2	battery	0.1148e+00	(0.6714e-02)	0.8645e-01	(0.1573e-01)	0.7988e+00	(0.1608e-01)
3	switch	0.1164e+00	(0.4372e-02)	0.	(0.)	0.8836e+00	(0.4372e-02)
4	wires	0.3213e-01	(0.2447e-02)	0.	(0.)	0.9679e+00	(0.2447e-02)

- effective availability of whole system = 64.1311 %

- sensitivity constants -

component ID	& name	lambda-21	lambda-20	lambda-10	tau-02	tau-01	tau-12	e-avail(appr.)
1	resistance	-0.2135e+00	-0.6406e+00	-0.4271e+00	-0.1281e+01	0.	0.	0.1215e+00
2	battery	-0.2495e+00	-0.3590e+00	-0.1095e+00	-0.7181e+00	0.	0.	0.3911e+00
3	switch	-0.1000e+01	0.	0.	0.	0.	0.	0.5000e+00
4	wires	-0.1000e+01	0.	0.	0.	0.	0.	0.5000e+00

- ranking of effective availability -

rank	gate ID	name	% eff-avail	error
1	2	power	67.3443	1.698775
2	1	system	64.1311	1.809820

rank	comp. ID	name	% eff-avail	error
1	4	wires	96.7869	0.484502
2	3	switch	88.3607	0.865712
3	2	battery	84.2022	1.878332
4	1	resistance	74.8579	1.407703

- ranking via importance -

rank	comp. ID	name	importance	error
1	1	resistance	0.1162e+00	0.6321e-02
2	2	battery	0.7626e-01	0.5188e-02
3	4	wires	0.3264e-01	0.2514e-02
4	3	switch	0.	0.

- **The maximum amount of matintenance equipments required at a time.**

equipment ID	name	amount	uncertainty
1	equipment 1	5.0	0.040
2	equipment 2	4.0	0.040

- The amount of spares required by this operation -

component ID	name	amount	uncertainty
3	switch	34.6	1.776

- computation of system cost in million dollars -

comp. ID	name	capital cost/unit	sch. maint. equip&person	**total cost**
1	resistance	10.00	5.00	15.00
2	battery	5.00	1.00	6.00
3	switch	55.20	12.00	2220.00
4	wires	10.00	0.00	10.00

sum for components = $ 2251.00

- economic data on repair facilities -

equip. ID	name	cost	total cost
1	equipment 1	25.00	125.00
2	equipment 2	25.00	100.00

sum for repair facility = $ 225.00

- total cost of this system = $ 2476.00

* computing time in seconds *
(cpu/i&o/sys)= 2.31305/ **0.01210/ 0.00220/**