

# Probability and Statistics with Reliability, Queuing and Computer Science Applications

Second edition

by K.S. Trivedi

Publisher-John Wiley & Sons

Chapter 8 (Part 5) :Continuous Time Markov Chains

## **Reliability Modeling**

Dept. of Electrical & Computer engineering

Duke University

Email:[kst@ee.duke.edu](mailto:kst@ee.duke.edu)

URL: [www.ee.duke.edu/~kst](http://www.ee.duke.edu/~kst)

# Outline of This Part of Chapter 8

- Hardware Reliability Models
- A Safety Model
- A Security Model
- A Real-Time System Model
- Software Reliability Growth Models

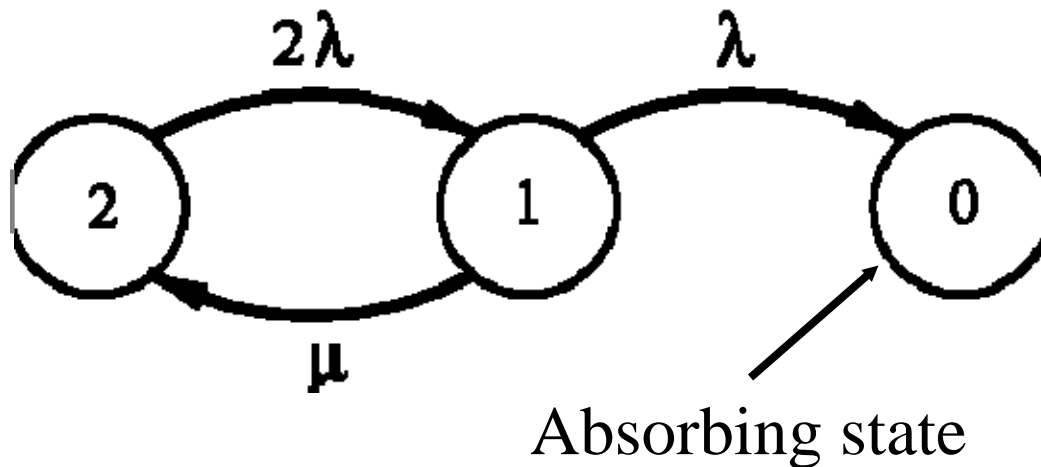
# Hardware Reliability Models

- Two component Markov reliability model with repair
- Two component Markov model with imperfect fault coverage
- WFS reliability model

# Markov Reliability Model With Repair

- Consider the 2-component parallel system (no delay + perfect coverage) but disallow repair from system down state.
- Note that state 0 is now an absorbing state. The state diagram is given in the following figure.
- This reliability model with repair cannot be modeled using a reliability block diagram or a fault tree. We need to resort to Markov chains. (This is a form of dependency since in order to repair a component you need to know the status of the other component).

# Markov Reliability Model With Repair (Contd.)



- Markov chain has an absorbing state. In the steady-state, system will be in state 0 with probability 1. Hence steady state analysis will yield a trivial answer; transient analysis is of interest. States 1 and 2 are transient states.

# Markov Reliability Model With Repair (Contd.)

- Some authors erroneously claim that reliability models do not admit repair.
- In the model on previous slide, we have component repair from state 1; system has not failed in this state.
- In a reliability model we do not allow repair from system failure states (such as state 0).
- Thus, there must be one or more absorbing states in a reliability model

# Markov Reliability Model With Repair (Contd.)

- Assume that the initial state of the Markov chain is 2, that is,  $\pi_2(0) = 1$ ,  $\pi_k(0) = 0$  for  $k = 0, 1$ .
- Then the system of differential Equations is written based on:

Rate of buildup = Rate of flow in - Rate of flow out  
for each state

## Markov Reliability Model With Repair (Contd.)

$$\frac{d\pi_2(t)}{dt} = -2\lambda\pi_2(t) + \mu\pi_1(t)$$

$$\frac{d\pi_1(t)}{dt} = 2\lambda\pi_2(t) - (\lambda + \mu)\pi_1(t)$$

$$\frac{d\pi_0(t)}{dt} = \lambda\pi_1(t)$$



# Markov Reliability Model With Repair (Contd.)

Using the technique of Laplace transform, we can reduce the above system to:

$$s\overline{\pi_2(s)} - 1 = -2\lambda\overline{\pi_2(s)} + \mu\overline{\pi_1(s)}$$

$$s\overline{\pi_1(s)} = 2\lambda\overline{\pi_2(s)} - (\lambda + \mu)\overline{\pi_1(s)}$$

$$s\overline{\pi_0(s)} = \lambda\overline{\pi_1(s)} \quad \text{where } \overline{\pi(s)} = \int_0^{\infty} e^{-st} \pi(t) dt$$

# Markov Reliability Model With Repair (Contd.)

Solving for  $\overline{\pi_0(s)}$ , we get:

$$\overline{\pi_0(s)} = \frac{2\lambda^2}{s[s^2 + (3\lambda + \mu)s + 2\lambda^2]}$$

- After an inversion, we can obtain  $\pi_0(t)$ , the probability that no components are operating at time  $t \geq 0$ . For this purpose, we carry out a partial fraction expansion.

# Markov Reliability Model With Repair (Contd.)

Inverting the transform, we get

$$R(t) = 1 - \pi_0(t) = \frac{2\lambda^2}{\alpha_1 - \alpha_2} \left( \frac{e^{-\alpha_1 t}}{\alpha_1} - \frac{e^{-\alpha_2 t}}{\alpha_2} \right)$$

where

$$\alpha_1, \alpha_2 = \frac{(3\lambda + \mu) \pm \sqrt{\lambda^2 + 6\lambda\mu + \mu^2}}{2}$$

# Markov Reliability Model With Repair (Contd.)

Recalling that  $MTTF = \int_0^{\infty} R(t) dt$ , we get:

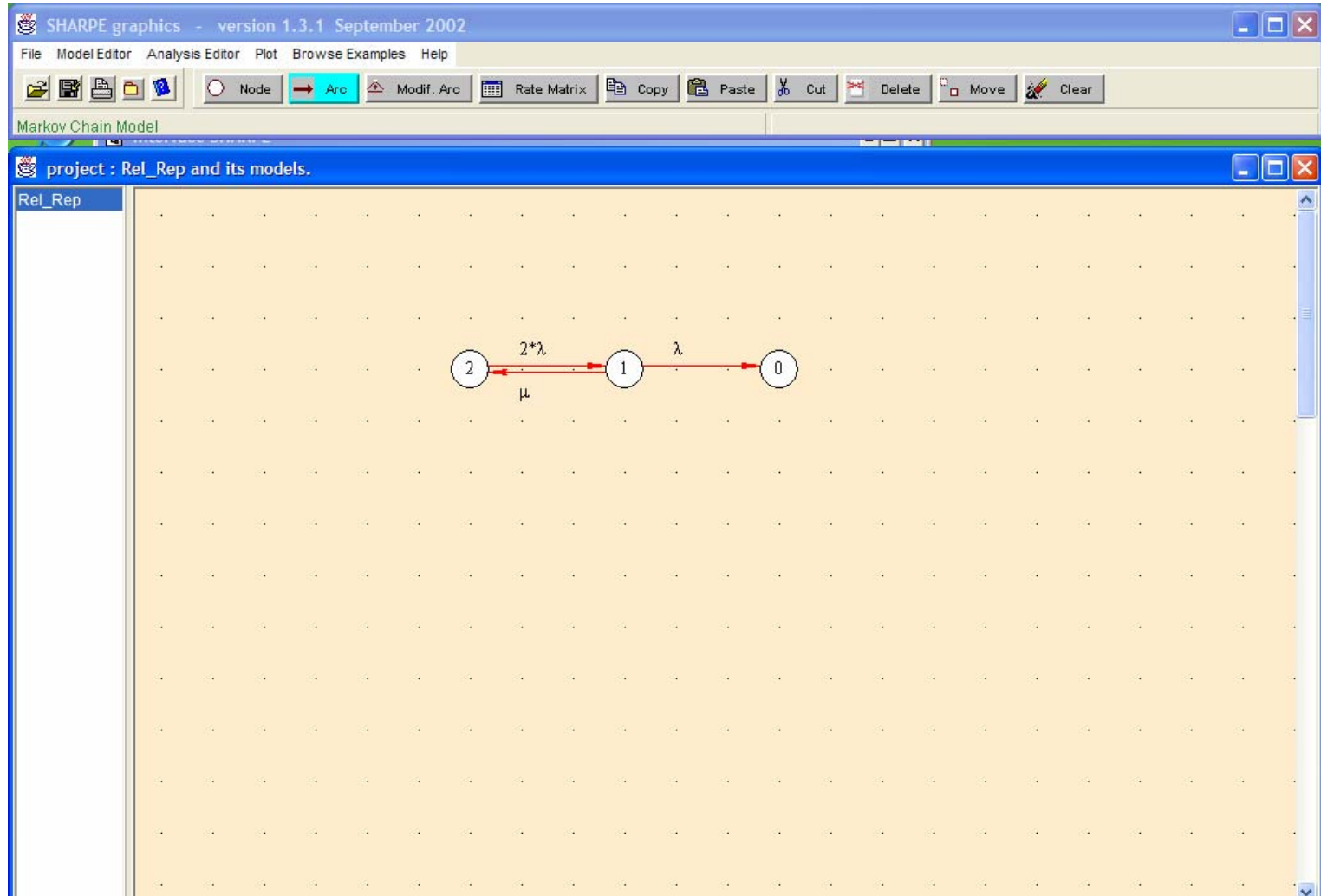
$$\begin{aligned} MTTF &= \frac{2\lambda^2}{\alpha_1 - \alpha_2} \left[ \frac{1}{\alpha_2^2} - \frac{1}{\alpha_1^2} \right] = \frac{2\lambda^2(\alpha_1 + \alpha_2)}{\alpha_1^2 \alpha_2^2} \\ &= \frac{2\lambda^2(3\lambda + \mu)}{(2\lambda^2)^2} = \frac{3}{2\lambda} + \frac{\mu}{2\lambda^2} \end{aligned}$$

# Markov Reliability Model With Repair (Contd.)

- Note that the MTTF of the two component parallel redundant system, in the absence of a repair facility (i.e.,  $\mu = 0$ ), would have been equal to the first term,  $3 / (2 * \lambda)$ , in the above expression.
- Therefore, the effect of a repair facility is to increase the mean life by  $\mu / (2 * \lambda^2)$ , or by a factor

$$\frac{\frac{\mu}{2\lambda^2}}{\frac{3}{2\lambda}} = \frac{\mu}{3\lambda}$$

# Model made in SHARPE GUI



# Parameters entered for the Model

SHARPE graphics - version 1.3.1 September 2002

File Model Editor Analysis Editor Plot Browse Examples Help

Node Arc Modif. Arc Rate Matrix Copy Paste Cut Delete Move Clear

Model with ABSOR... Analysis frame: Parameters Code Output Graph Personal Modification

project : Rel\_Rep

Rel\_Rep

Name of the model: Rel\_Rep

Output(s) selected:  
Reliability  
Mean time to absorption (failure)

Outputs:

- Cumulative dist. function of time to absorption
- Reliability
- Prob of absorption at time t (unreliability)
- Mean time to absorption (failure)
- State probability at absorption
- State probability at absorption for all states
- Expected reward rate at time t (ext)

Add

Delete

Parameter(s) for the current output:

Values for the variables not bound:

Variable	Value
lambda	0.0002
mu	1/5

Run

Close Help

# Sharpe Input file generated by GUI

- format 8
- factor on

Model defined

- markov Rel\_Rep(lambda, mu)
- 2 1 2\*lambda
- 1 0 lambda
- 1 2 mu
- end
- \* Initial Probabilities defined:
- 2 init\_Rel\_Rep\_2
- 1 init\_Rel\_Rep\_1
- 0 init\_Rel\_Rep\_0
- end

- \* Initial Probailities assigned:
- bind
- init\_Rel\_Rep\_2 0
- init\_Rel\_Rep\_1 0
- init\_Rel\_Rep\_0 0
- end

- echo  
\*\*\*\*\*  
\*\*\*\*\*
- echo \*\*\*\*\* Outputs asked for the model: Rel\_Rep  
\*\*\*\*\*

Output asked

- \* Initial Probability: ini1
- bind
- init\_Rel\_Rep\_2 1
- init\_Rel\_Rep\_1 0
- init\_Rel\_Rep\_0 0
- end

Initial prob. assigned

- bind lambda 0.0002
- bind mu 1/5

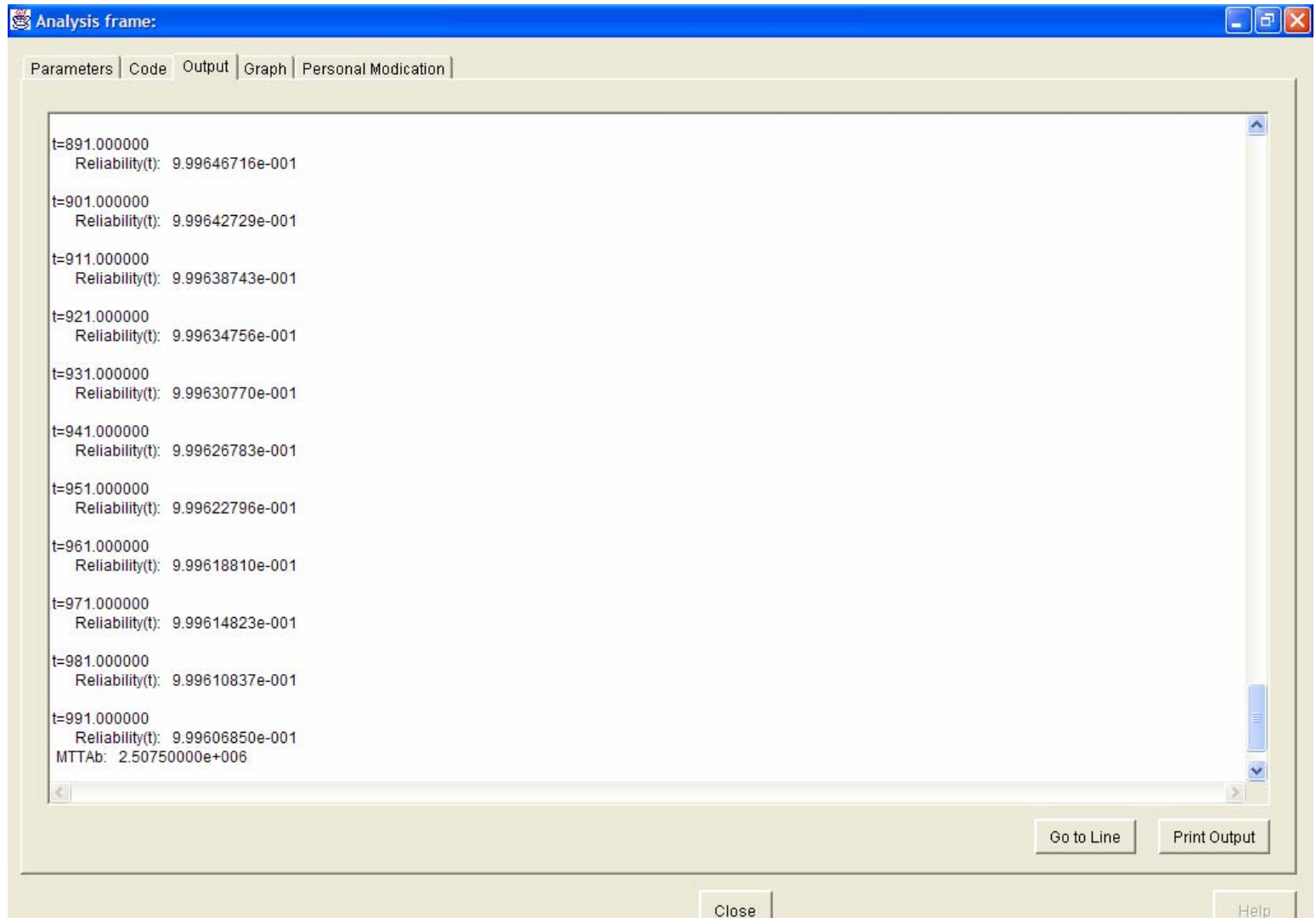
- func Reliability(t) 1-tvalue(t;Rel\_Rep; lambda, mu)
- loop t,1,1000,10
- expr Reliability(t)
- end

- bind lambda 0.0002
- bind mu 1/5

- var MTTAb mean(Rel\_Rep, 0; lambda, mu)
- expr MTTAb
- end



# Output generated by SHARPE GUI

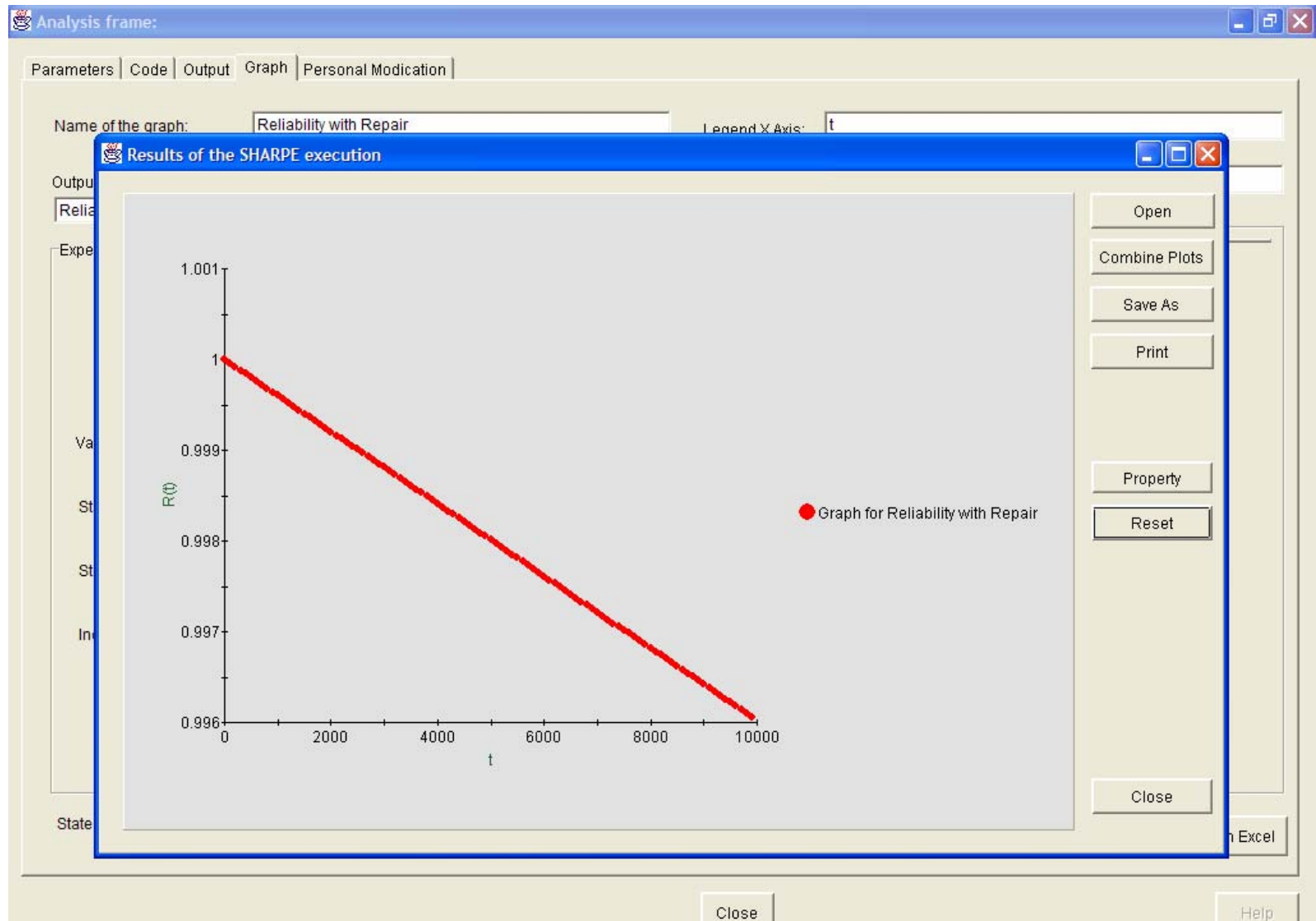


The screenshot shows a window titled "Analysis frame:" with a blue title bar and standard Windows window controls. Below the title bar is a menu bar with tabs for "Parameters", "Code", "Output", "Graph", and "Personal Modication". The "Output" tab is selected. The main area contains a text box with the following output:

```
t=891.000000  
Reliability(t): 9.99646716e-001  
  
t=901.000000  
Reliability(t): 9.99642729e-001  
  
t=911.000000  
Reliability(t): 9.99638743e-001  
  
t=921.000000  
Reliability(t): 9.99634756e-001  
  
t=931.000000  
Reliability(t): 9.99630770e-001  
  
t=941.000000  
Reliability(t): 9.99626783e-001  
  
t=951.000000  
Reliability(t): 9.99622796e-001  
  
t=961.000000  
Reliability(t): 9.99618810e-001  
  
t=971.000000  
Reliability(t): 9.99614823e-001  
  
t=981.000000  
Reliability(t): 9.99610837e-001  
  
t=991.000000  
Reliability(t): 9.99606850e-001  
MTTAb: 2.50750000e+006
```

At the bottom right of the window are buttons for "Go to Line" and "Print Output". At the very bottom of the window are buttons for "Close" and "Help".

# Graph between Reliability and time



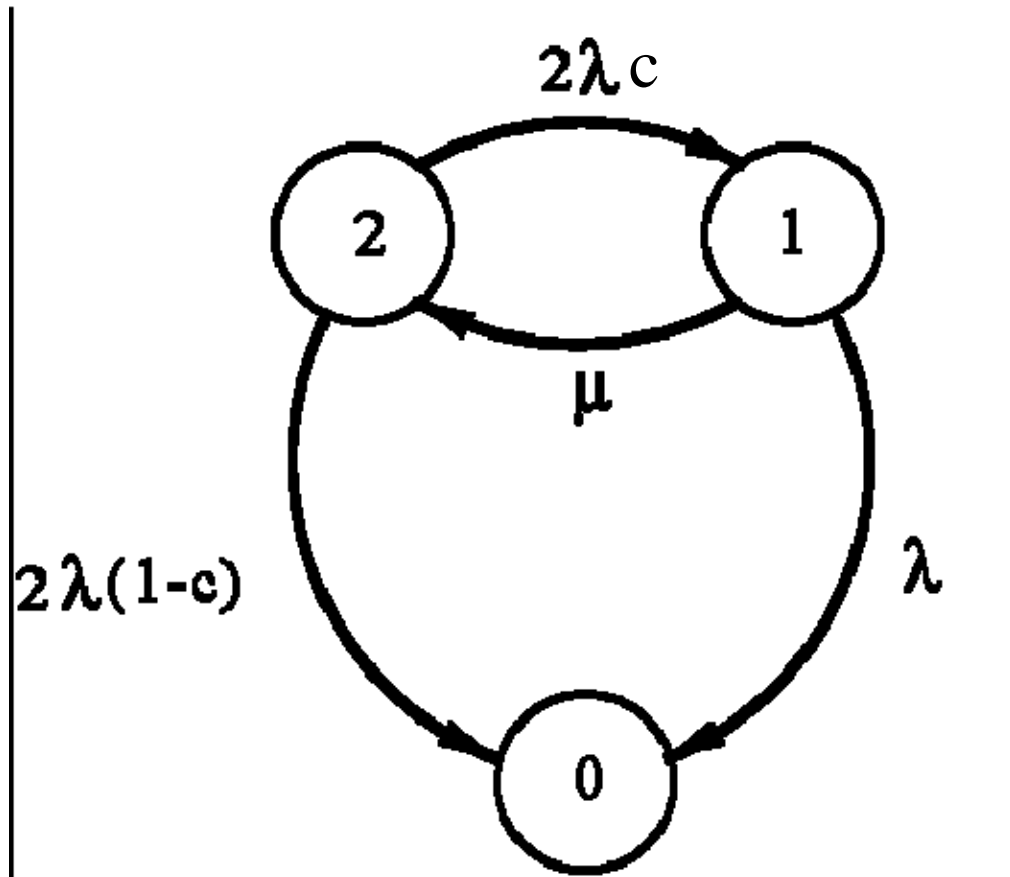
# Markov Reliability Model With Imperfect Coverage

# Markov Model With Imperfect Coverage

- Next consider a modification of the above example proposed by Arnold as a model of duplex processors of an electronic switching system.
- Assuming that not all faults are recoverable and that  $c$  is the coverage factor which denotes the conditional probability that the system recovers given that a fault has occurred.
- The state diagram is now given by the following picture:

# Markov Model With Imperfect Coverage

(Contd.)



# Markov Model With Imperfect Coverage (Contd.)

- Assume that the initial state is 2 so that:

$$\pi_2(0) = 1, \quad \pi_0(0) = \pi_1(0) = 0$$

- Then the system of differential equations are:

$$\frac{d\pi_2(t)}{dt} = -2\lambda c \pi_2(t) - 2\lambda(1-c) \pi_2(t) + \mu \pi_1(t)$$

$$\frac{d\pi_1(t)}{dt} = 2\lambda c \pi_2(t) - (\lambda + \mu) \pi_1(t)$$

$$\frac{d\pi_0(t)}{dt} = 2\lambda(1-c) \pi_2(t) + \lambda \pi_1(t)$$

# Markov Model With Imperfect Coverage (Contd.)

Using Laplace transforms as before, the above system reduces to:

$$s\overline{\pi_2(s)} - 1 = -2\lambda\overline{\pi_2(s)} + \mu\overline{\pi_1(s)}$$

$$s\overline{\pi_1(s)} = 2\lambda c\overline{\pi_2(s)} - (\lambda + \mu)\overline{\pi_1(s)}$$

$$s\overline{\pi_0(s)} = \lambda\overline{\pi_1(s)} + 2\lambda(1-c)\overline{\pi_2(s)}$$

# Markov Model With Imperfect Coverage (Contd.)

- After solving the differential equations we obtain:

$$R(t) = \pi_2(t) + \pi_1(t)$$

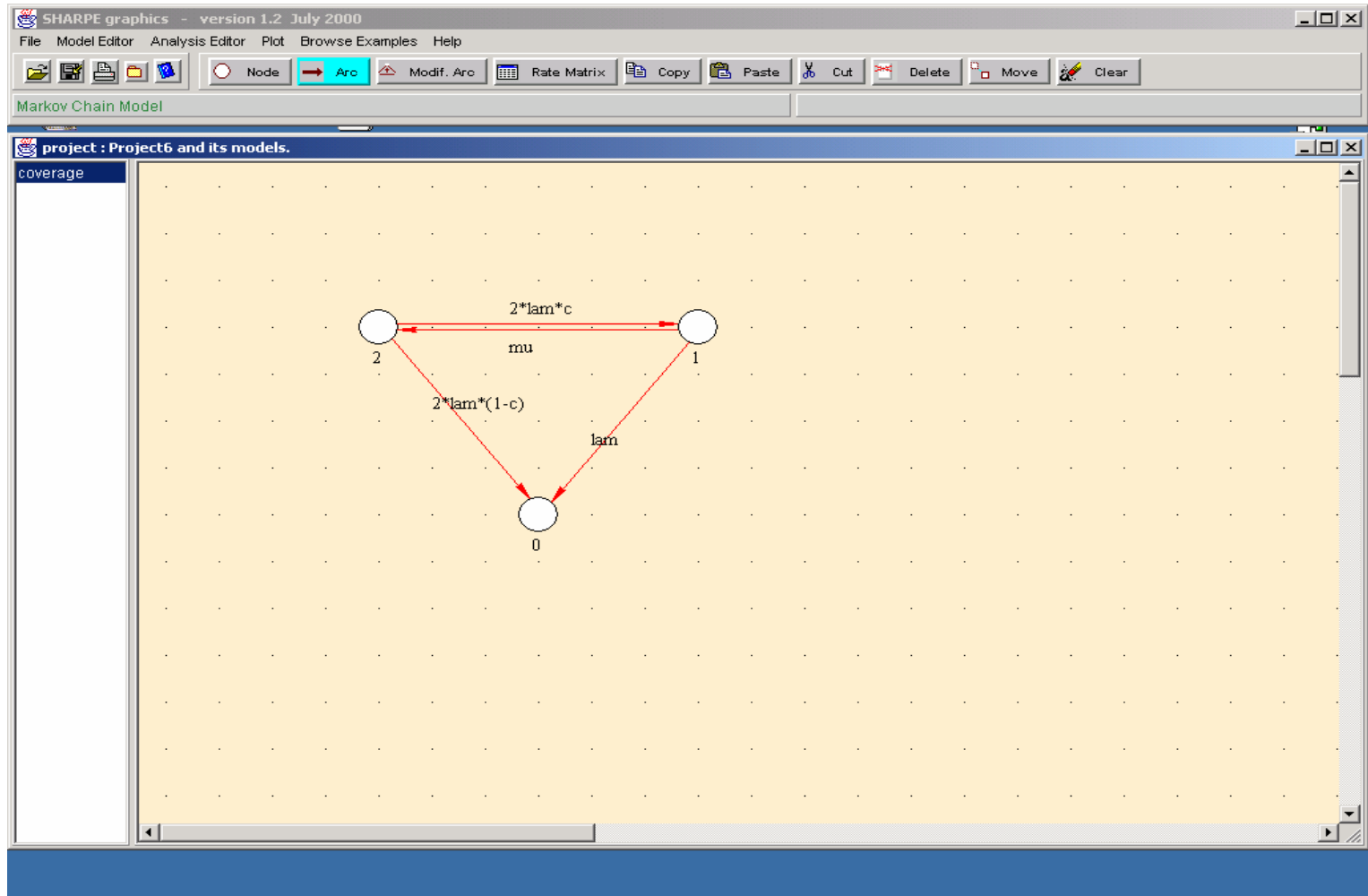
- From  $R(t)$ , we can system  $MTTF$ :

$$MTTF = \frac{\lambda(1 + 2c) + \mu}{2\lambda[\lambda + \mu(1 - c)]}$$

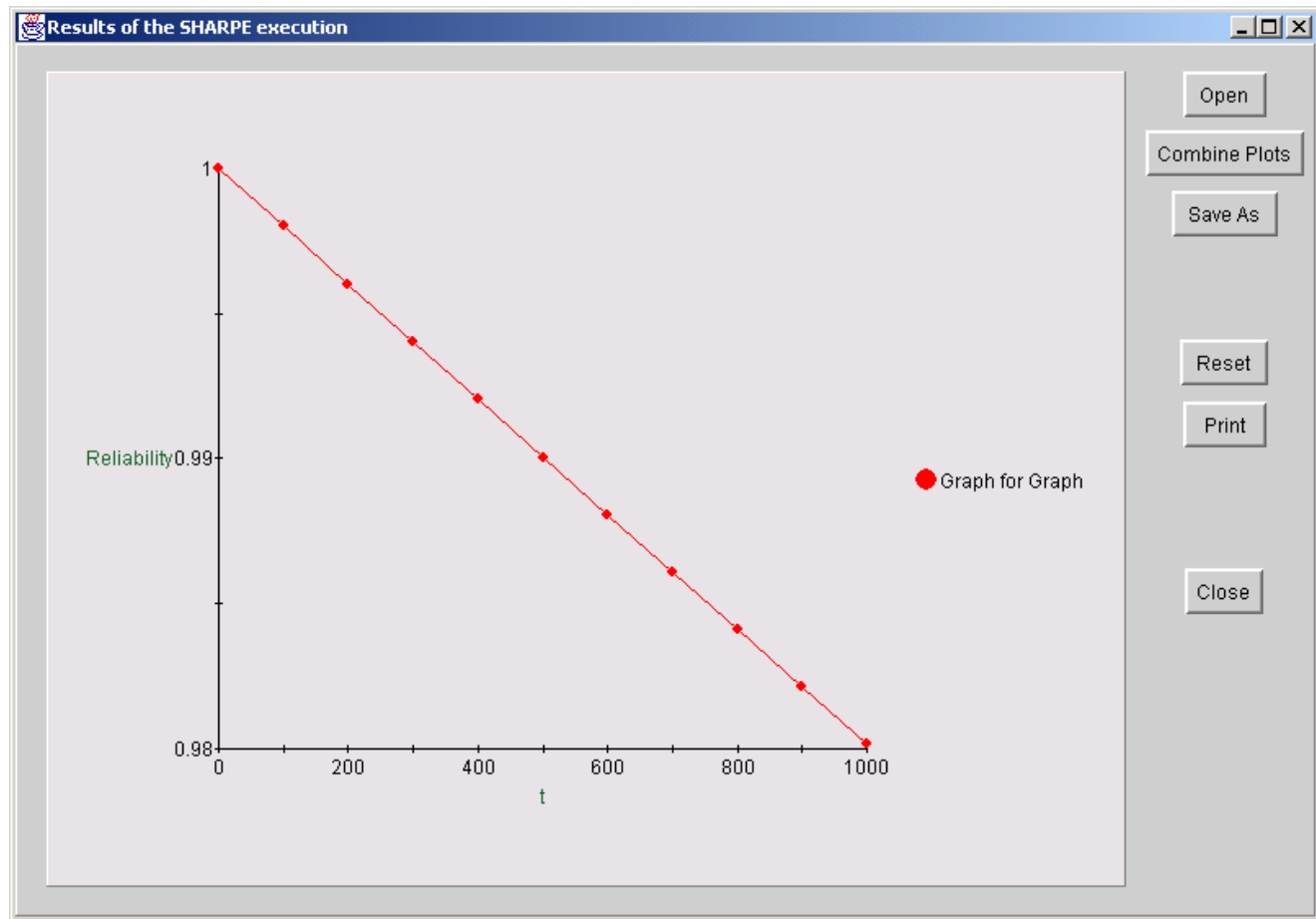
- It should be clear that the system  $MTTF$  and system reliability are critically dependent on the coverage factor.



# Model made in SHARPE GUI



# Graph between $R(t)$ and time

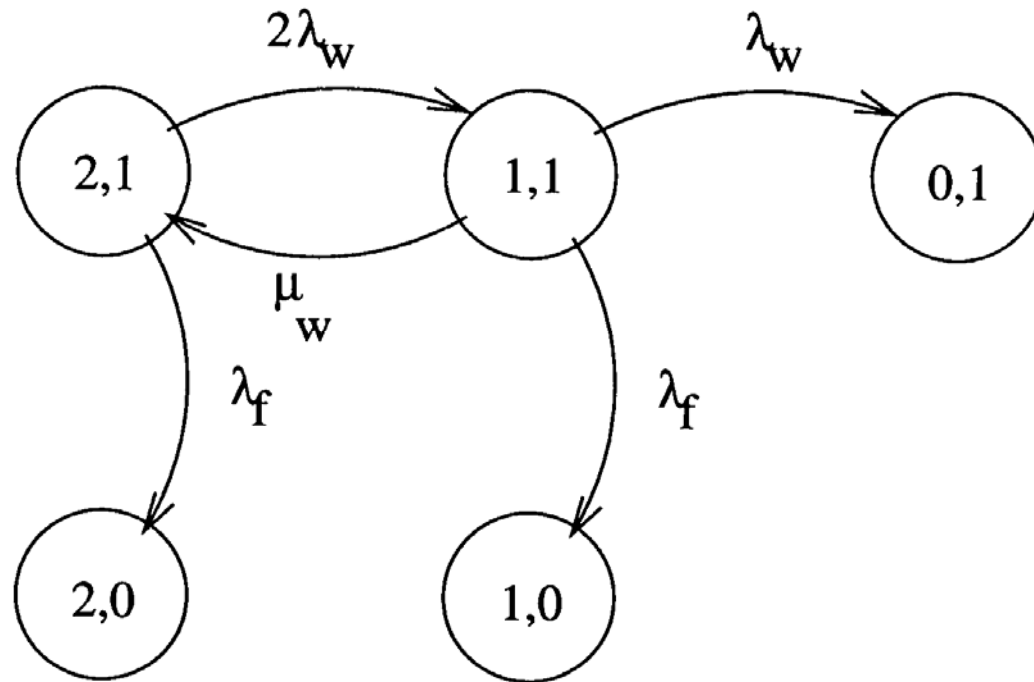


# Markov Reliability Model with Repair (WFS Example)

# Markov Reliability Model With Repair (WFS Example)

- WFS: Workstation File System
- Assume that the computer system does not recover if both workstations fail, or if the file-server fails.

# Markov Reliability Model With Repair



- States (0,1), (1,0) and (2,0) become absorbing states while (2,1) and (1,1) are transient states.
- **Note:** we have made a simplification that, once the CTMC reaches a system failure state, we do not allow any more transitions.

## Markov Reliability Model With Repair (Contd.)

- If we solve for  $\pi_{2,1}(t)$  and  $\pi_{1,1}(t)$  then

$$R(t) = \pi_{2,1}(t) + \pi_{1,1}(t)$$

- For a Markov chain with absorbing states:

$A$ : the set of absorbing states

$B = \Omega - A$ : the set of remaining states

$\tau_{i,j}$ : Mean time spent in state  $i,j$  until absorption

$$\tau_{i,j} = \int_0^{\infty} \pi_{i,j}(x) dx, \quad (i, j) \in B$$

$$\tau Q_B = -\pi_B(0)$$

# Markov Reliability Model With Repair (Contd.)

- $Q_B$  derived from  $Q$  by restricting it to only states in  $B$
- Mean time to absorption  $MTTA$  is given as:

$$MTTA = \sum_{(i,j) \in B} \tau_{i,j}$$

## Markov Reliability Model With Repair (Contd.)

$$Q_B = \begin{bmatrix} -(\lambda_f + 2\lambda_w) & 2\lambda_w \\ \mu_w & -(\mu_w + \lambda_f + \lambda_w) \end{bmatrix}$$

First solve

$$\frac{d\pi_{2,1}(t)}{dt} = -(2\lambda_w + \lambda_f)\pi_{2,1}(t) + \mu_w\pi_{1,1}(t)$$
$$\frac{d\pi_{1,1}(t)}{dt} = -(\mu_w + \lambda_f + \lambda_w)\pi_{1,1}(t) + 2\lambda_w\pi_{2,1}(t)$$



## Markov Reliability Model With Repair (Contd.)

Then :  $R(t) = \pi_{2,1}(t) + \pi_{1,1}(t)$

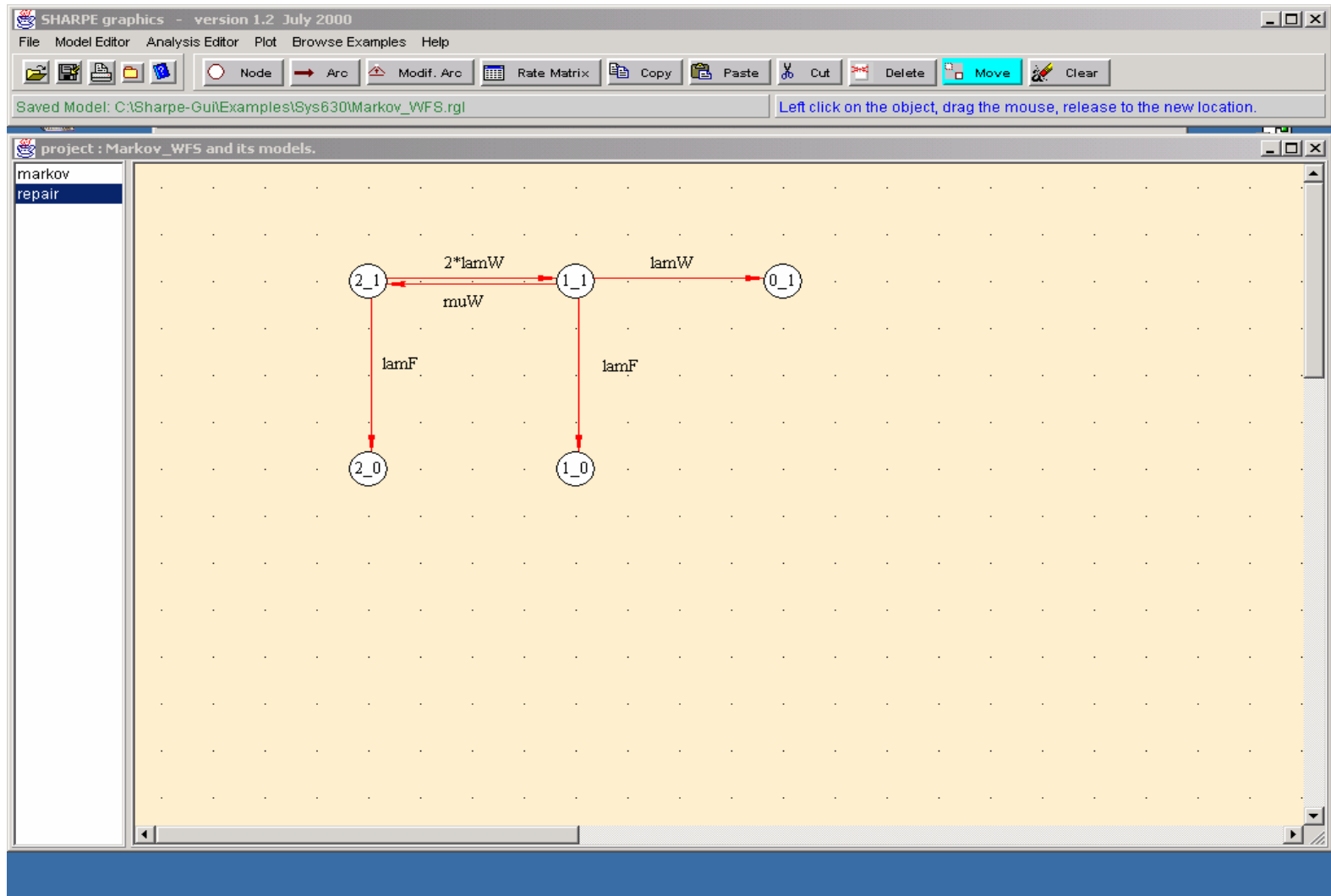
next solve  $\tau_{2,1}(-(\lambda_f + 2\lambda_w)) + \tau_{1,1}\mu_w = -1$

$$\tau_{2,1} 2\lambda_w - \tau_{1,1}(\mu_w + \lambda_f + \lambda_w) = 0$$

Then :  $MTTF = \tau_{2,1} + \tau_{1,1}$

- Mean time to failure is 19992 hours (input values refer to Part 2 of Chapter 8).

# Model made in SHARPE GUI



# Parameters assigned and output asked

The screenshot shows the SHARPE graphics software interface. The main window is titled "SHARPE graphics - version 1.3.1 September 2002". The menu bar includes "File", "Model Editor", "Analysis Editor", "Plot", "Browse Examples", and "Help". The toolbar contains icons for "Node", "Arc", "Modif. Arc", "Rate Matrix", "Copy", "Paste", "Cut", "Delete", "Move", and "Clear".

The "Analysis frame" is open, showing the "Parameters" tab. The "Name of the model:" dropdown is set to "repair". The "Outputs:" list includes "Cumulative dist. function of time to absorption", "Reliability", "Prob of absorption at time t (unreliability)", "Mean time to absorption (failure)", "State probability at absorption", "State probability at absorption for all states", and "Expected reward rate at time t (ext)". The "Output(s) selected:" list includes "Mean time to absorption (failure)" and "Reliability".

The "Parameter(s) for the current output:" section is empty. The "Values for the variables not bound:" table is shown below:

Variable	Value
lamW	0.0003
lamF	0.0001
muW	1

The "Run" button is highlighted in yellow. Other buttons include "Close" and "Help".

# SHARPE (textual) input file

Model defined

- format 8
- factor on
  
- markov repair(lamW, lamF, muW)
- 2\_1 1\_1 2\*lamW
- 2\_1 2\_0 lamF
- 1\_1 0\_1 lamW
- 1\_1 1\_0 lamF
- 1\_1 2\_1 muW
- end
- \* Initial Probabilities defined:
- 2\_1 init\_repair\_2\_1
- 1\_1 init\_repair\_1\_1
- 0\_1 init\_repair\_0\_1
- 2\_0 init\_repair\_2\_0
- 1\_0 init\_repair\_1\_0
- end
  
- \* Initial Probailities assigned:
- bind
- init\_repair\_2\_1 0
- init\_repair\_1\_1 0
- init\_repair\_0\_1 0
- init\_repair\_2\_0 0
- init\_repair\_1\_0 0
- end

- echo  
\*\*\*\*\*
- echo \*\*\*\*\* Outputs asked for the model: repair \*\*\*\*\*
  
- \* Initial Probability: config1
- bind
- init\_repair\_1\_0 0
- init\_repair\_0\_1 0
- init\_repair\_2\_1 1
- init\_repair\_2\_0 0
- init\_repair\_1\_1 0
- end
  
- bind lamW 0.0003
- bind lamF 0.0001
- bind muW 1
  
- var MTTAb mean(repair; lamW, lamF, muW)
- echo Mean time to absorption for repair
- expr MTTAb
  
- bind lamW 0.0003
- bind lamF 0.0001
- bind muW 1
  
- func Reliability(t) 1-tvalue(t;repair; lamW, lamF, muW)
- loop t,1,1000,100
- expr Reliability(t)
- end
  
- end

Output asked

Initial prob. assigned

# Output generated by SHARPE GUI

Analysis frame: [min] [max] [close]

Parameters | Code | **Output** | Graph | Personal Modication

```
***** Outputs asked for the model: repair *****
Input parameters values: lamW= 0.0003, lamF=0.0001, muW=1
Output:
Mean time to absorption for repair
MTTAb: 9.98205026e+003

-----

t=1.000000
  Reliability(t): 9.99899939e-001

t=101.000000
  Reliability(t): 9.89933033e-001

t=201.000000
  Reliability(t): 9.80065415e-001

t=301.000000
  Reliability(t): 9.70296159e-001

t=401.000000
  Reliability(t): 9.60624285e-001

t=501.000000
  Reliability(t): 9.51048821e-001

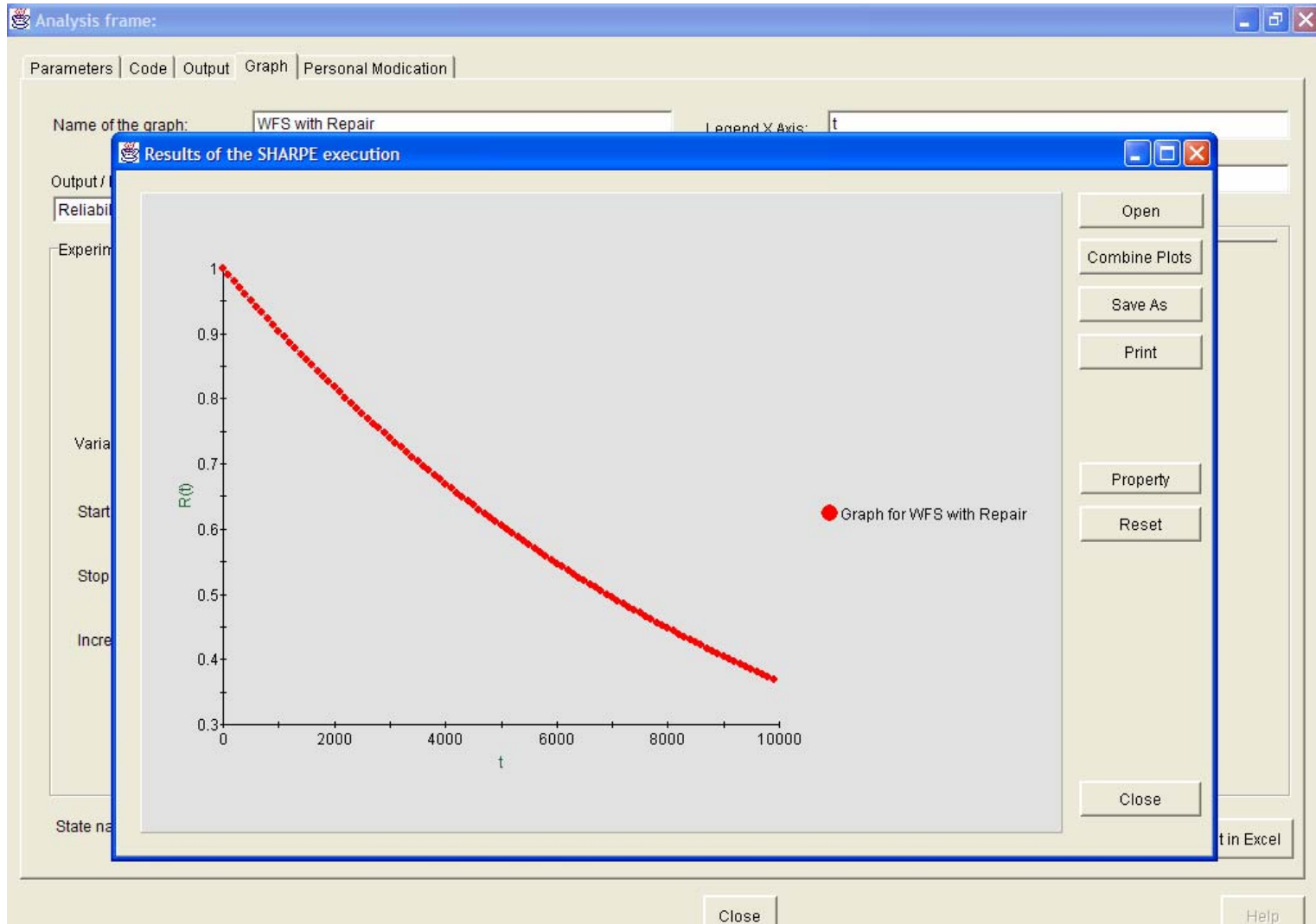
t=601.000000
  Reliability(t): 9.41568804e-001

t=701.000000
```

Go to Line    Print Output

Close    Help

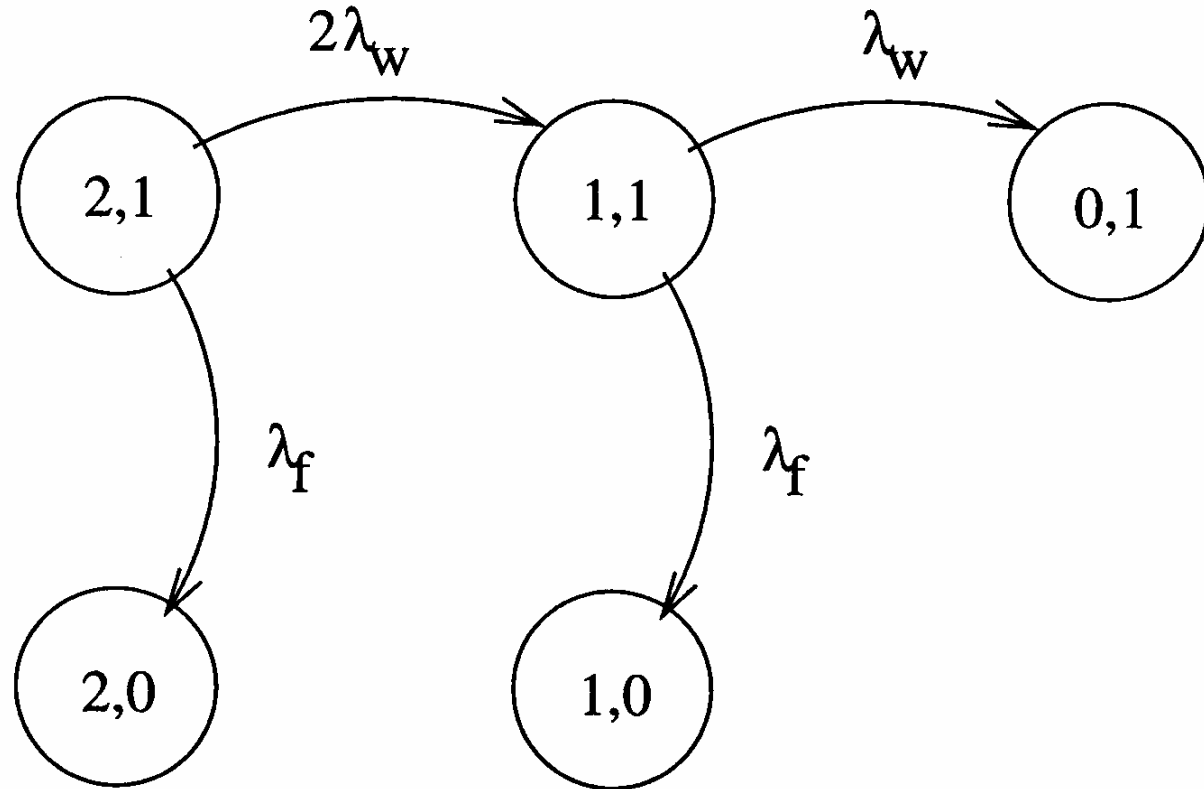
# Graph between $R(t)$ and time



# Markov Reliability Model Without Repair

# Markov Reliability Model without Repair: Case 1

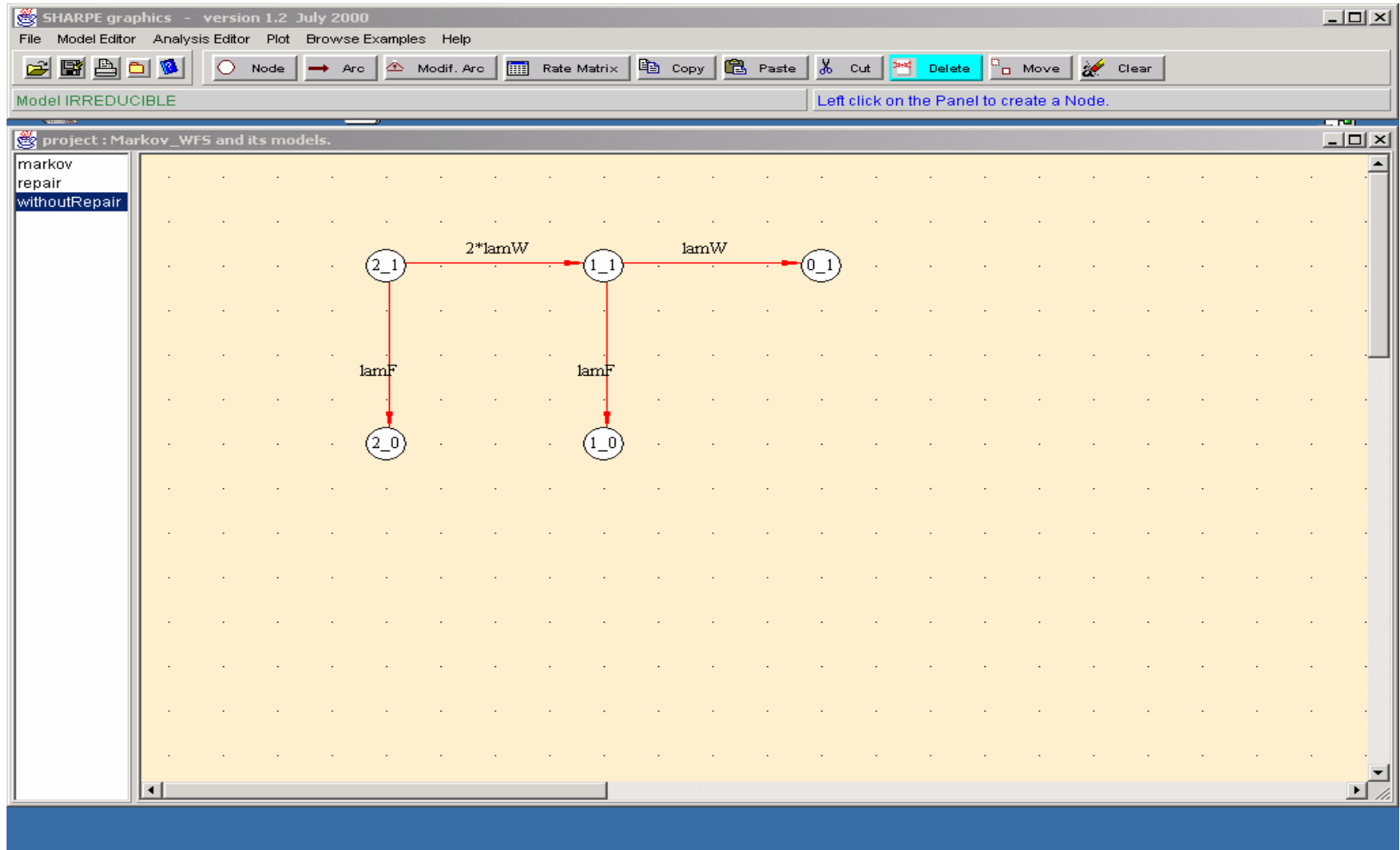
(Contd.)



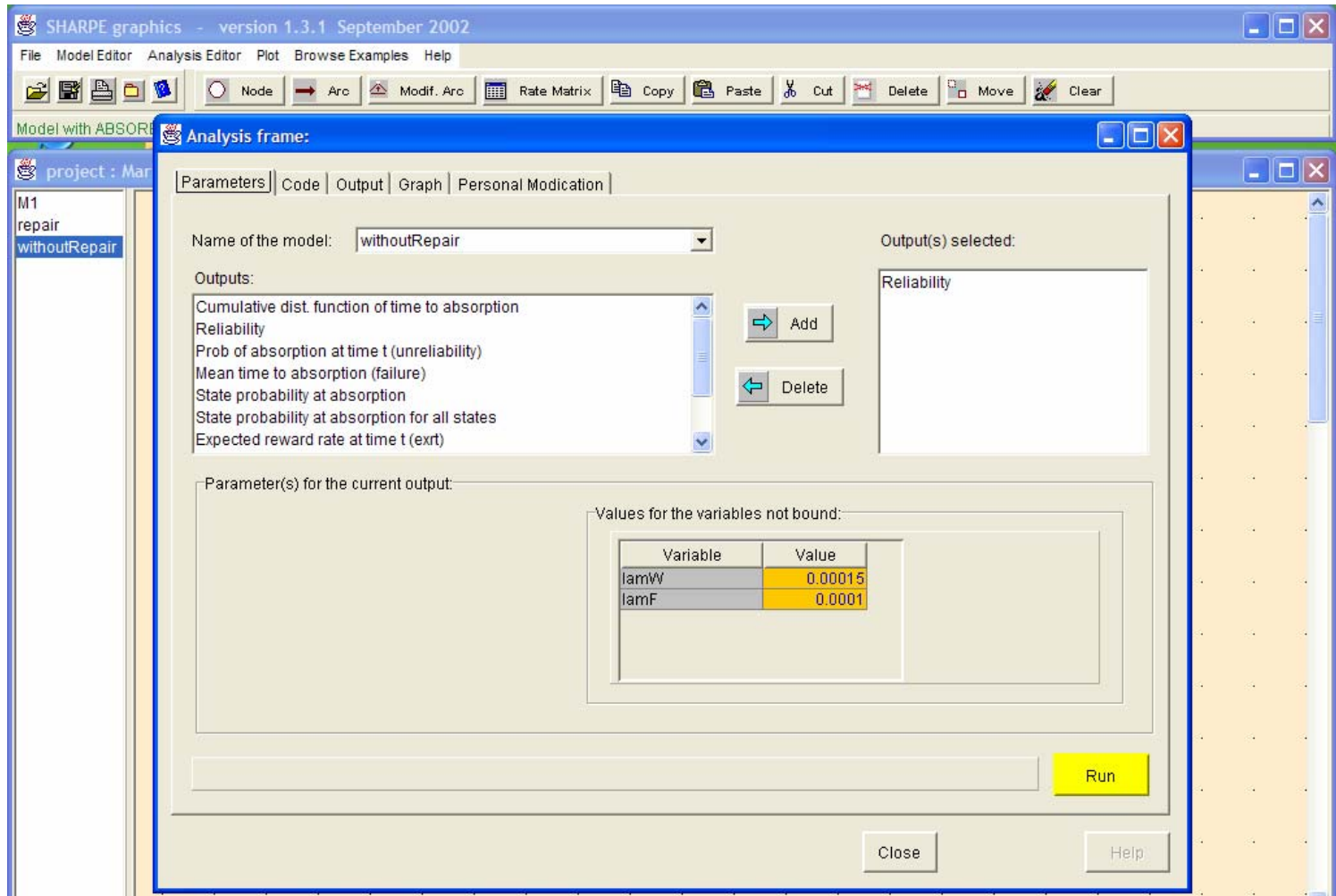
States (0,1), (1,0) and (2,0)  
become absorbing states



# Model made in SHARPE GUI



# Parameters assigned and Output asked



# Output generated by SHARPE GUI

```
***** Outputs asked for the model: withoutRepair *****
Input parameters values: lamW= 0.00015, lamF=0.0001
Output:

t=1.000000
  Reliability(t): 9.99899983e-001

t=101.000000
  Reliability(t): 9.89727030e-001

t=201.000000
  Reliability(t): 9.79236120e-001

t=301.000000
  Reliability(t): 9.68457419e-001

t=401.000000
  Reliability(t): 9.57419474e-001

t=501.000000
  Reliability(t): 9.46149286e-001

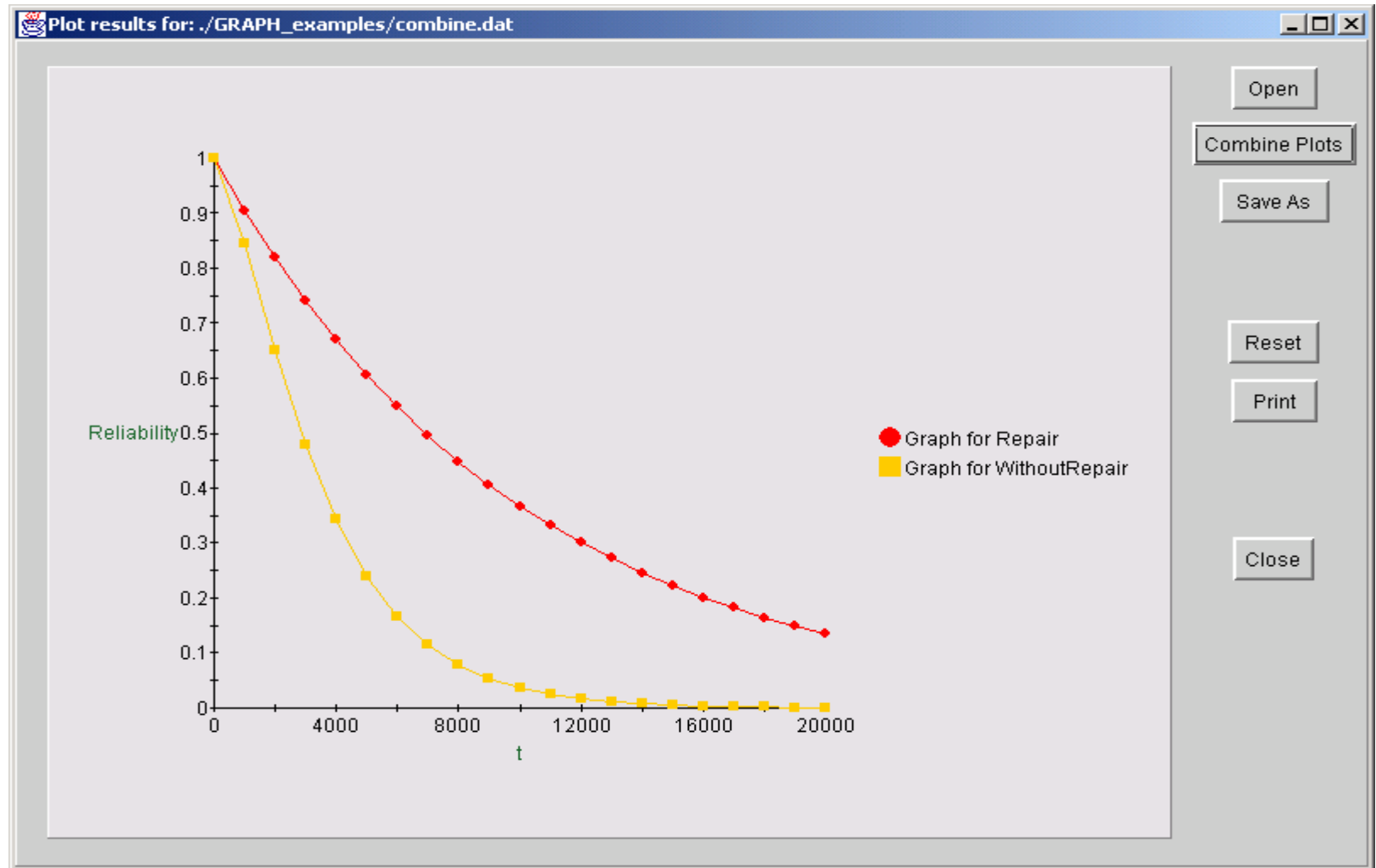
t=601.000000
  Reliability(t): 9.34672380e-001

t=701.000000
  Reliability(t): 9.23012877e-001

t=801.000000
  Reliability(t): 9.11193557e-001

t=901.000000
  Reliability(t): 8.99235918e-001
```

# Overlapped graph R(t) for with and without repair



# Markov Reliability Model without Repair: Case 1

(Contd.)

$$Q_B = \begin{bmatrix} -(\lambda_f + 2\lambda_w) & 2\lambda_w \\ 0 & -(\lambda_f + \lambda_w) \end{bmatrix}$$

$$R(t) = \pi_{2,1}(t) + \pi_{1,1}(t)$$

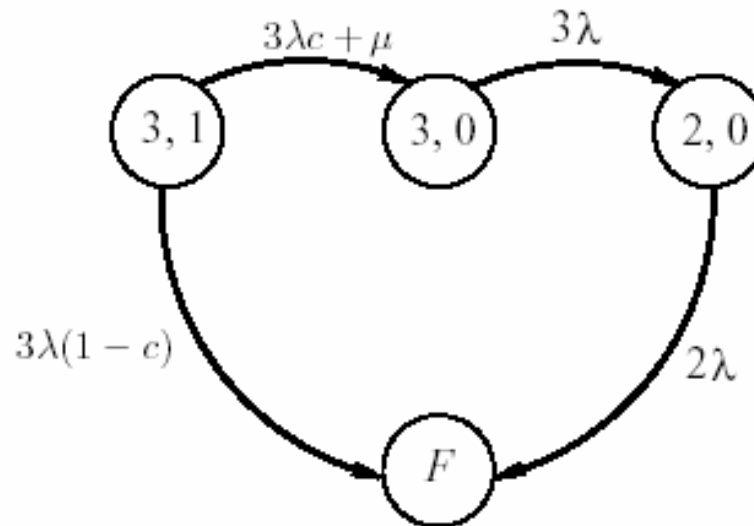
$$MTTF = \tau_{2,1} + \tau_{1,1}$$

- Mean time to failure is 9333 hours (see Part2 of Chapter 8).

# 3 Active Units and One Spare

## 3 Active Units and One Spare

- Consider a system with three active units and one spare. The active configuration is operated in TMR (Triple Modular Redundancy) mode. An active unit has a failure rate  $\lambda$ , while a standby spare unit has a failure rate  $\mu$ .



### 3 Active Units and One Spare (Contd.)

- Differential equations for this CTMC are written as follows:

$$\begin{aligned}\frac{d\pi_{3,1}}{dt} &= -(3\lambda + \mu)\pi_{3,1}(t), \\ \frac{d\pi_{3,0}}{dt} &= -3\lambda\pi_{3,0}(t) + (3\lambda c + \mu)\pi_{3,1}(t), \\ \frac{d\pi_{2,0}}{dt} &= -2\lambda\pi_{2,0}(t) + 3\lambda\pi_{3,0}(t), \\ \frac{d\pi_F}{dt} &= 3\lambda(1 - c)\pi_{3,1}(t) + 2\lambda\pi_{2,0}(t),\end{aligned}$$

Solving this system of equations, we get

$$\begin{aligned}\bar{\pi}_{3,1}(s) &= \frac{1}{s + 3\lambda + \mu}, \\ \bar{\pi}_{3,0}(s) &= \frac{3\lambda c + \mu}{(s + 3\lambda + \mu)(s + 3\lambda)}, \\ \bar{\pi}_{2,0}(s) &= \frac{3\lambda(3\lambda c + \mu)}{(s + 3\lambda + \mu)(s + 3\lambda)(s + 2\lambda)},\end{aligned}$$



### 3 Active Units and One Spare (Contd.)

and

$$s\bar{n}_F(s) = \frac{3\lambda(1-c)}{(s+3\lambda+\mu)} + \frac{6\lambda^2(3\lambda c + \mu)}{(s+2\lambda)(s+3\lambda)(s+3\lambda+\mu)}.$$

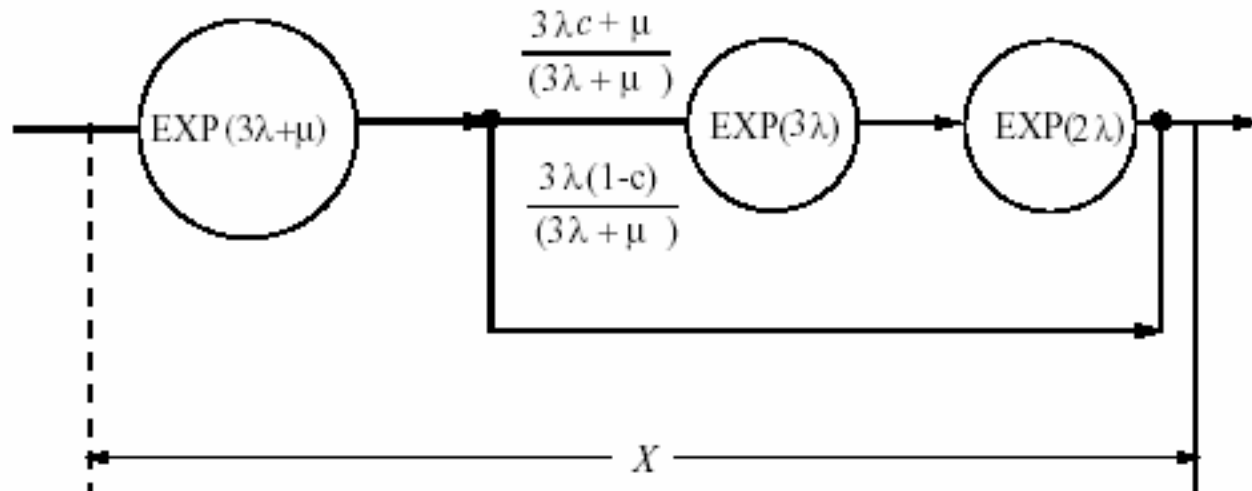
- So lifetime distribution becomes

$$\bar{f}_X(s) = \frac{3\lambda + \mu}{(s+3\lambda+\mu)} \left[ \frac{3\lambda(1-c)}{(3\lambda+\mu)} + \frac{3\lambda c + \mu}{3\lambda + \mu} \left\{ \frac{2\lambda}{(s+2\lambda)} + \frac{3\lambda}{(s+3\lambda)} \right\} \right]$$

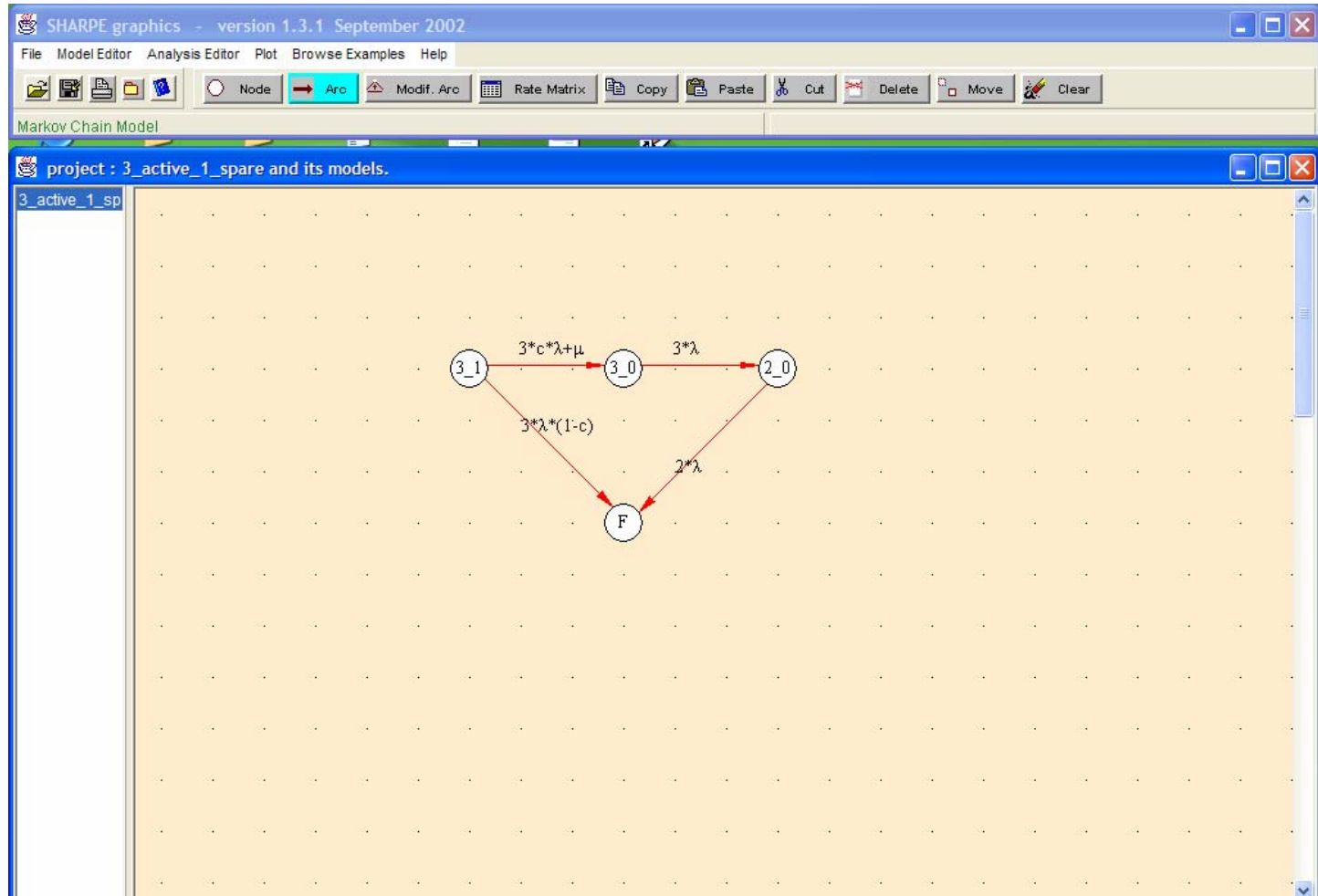
- The expression outside the square brackets is the Laplace–Stieltjes transform of  $\text{EXP}(3\lambda+\mu)$ , while the expression within the braces is the LST of  $\text{HYPO}(2\lambda, 3\lambda)$ .

### 3 Active Units and One Spare (contd.)

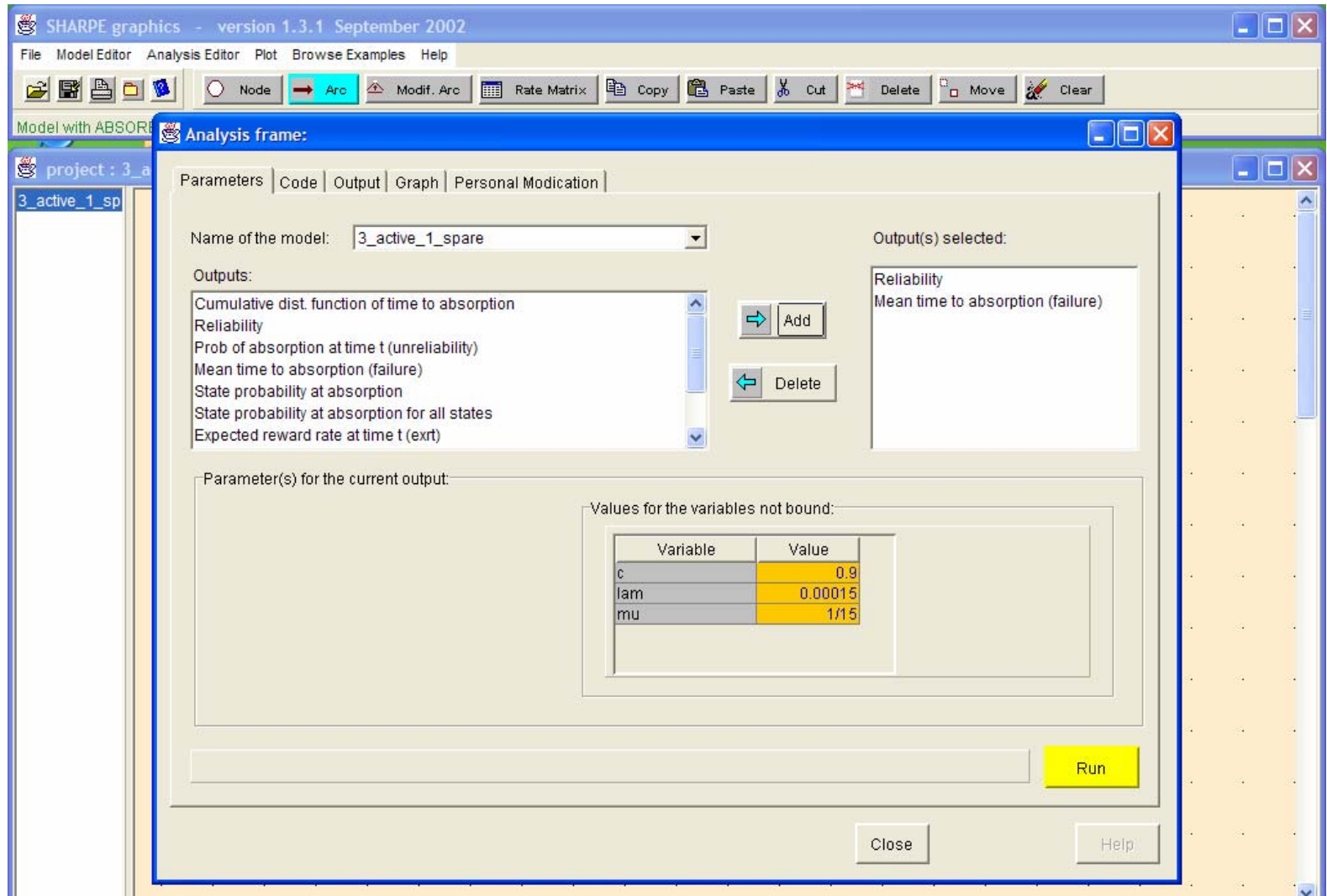
- Therefore, the system lifetime  $X$  has the stage-type distribution given as in this figure.



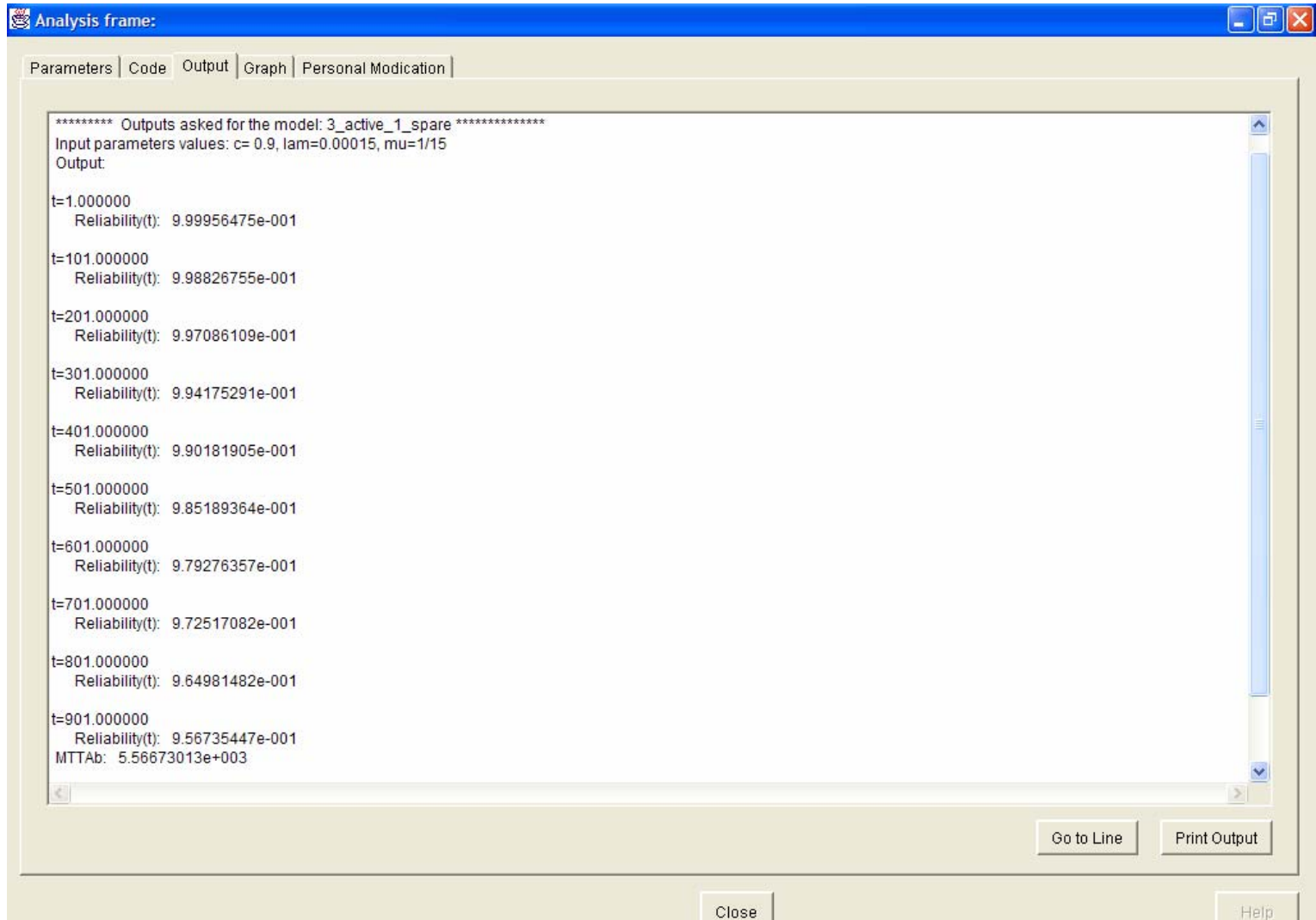
# Model made in SHARPE GUI



# Parameter assigned and output asked



# Output generated by SHARPE GUI

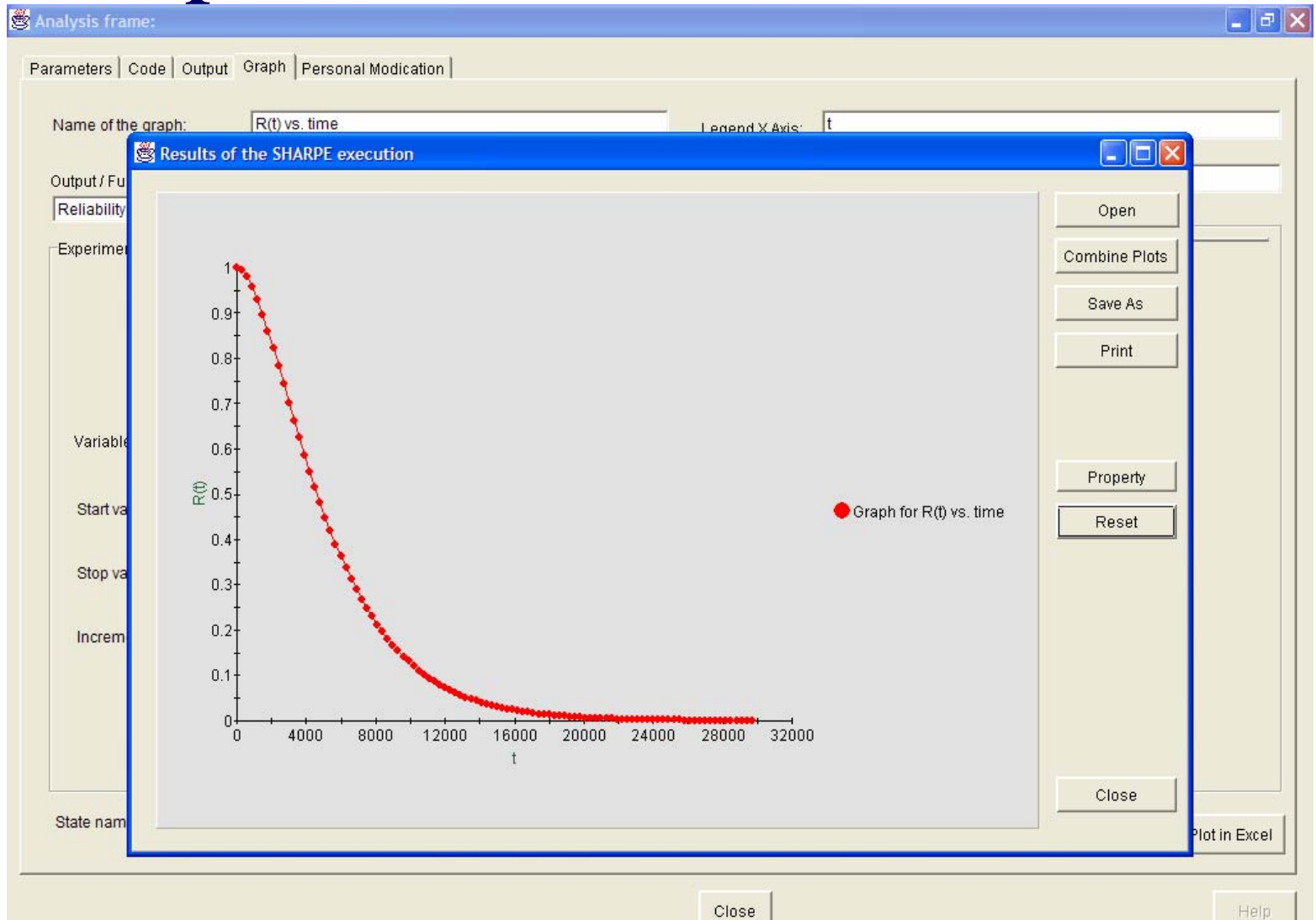


The screenshot shows a software window titled "Analysis frame:" with a blue title bar. Below the title bar are tabs for "Parameters", "Code", "Output", "Graph", and "Personal Modication". The "Output" tab is selected, displaying the following text:

```
***** Outputs asked for the model: 3_active_1_spare *****  
Input parameters values: c= 0.9, lam=0.00015, mu=1/15  
Output:  
  
t=1.000000  
  Reliability(t): 9.99956475e-001  
  
t=101.000000  
  Reliability(t): 9.98826755e-001  
  
t=201.000000  
  Reliability(t): 9.97086109e-001  
  
t=301.000000  
  Reliability(t): 9.94175291e-001  
  
t=401.000000  
  Reliability(t): 9.90181905e-001  
  
t=501.000000  
  Reliability(t): 9.85189364e-001  
  
t=601.000000  
  Reliability(t): 9.79276357e-001  
  
t=701.000000  
  Reliability(t): 9.72517082e-001  
  
t=801.000000  
  Reliability(t): 9.64981482e-001  
  
t=901.000000  
  Reliability(t): 9.56735447e-001  
MTTAb: 5.56673013e+003
```

At the bottom of the window, there are buttons for "Go to Line", "Print Output", "Close", and "Help".

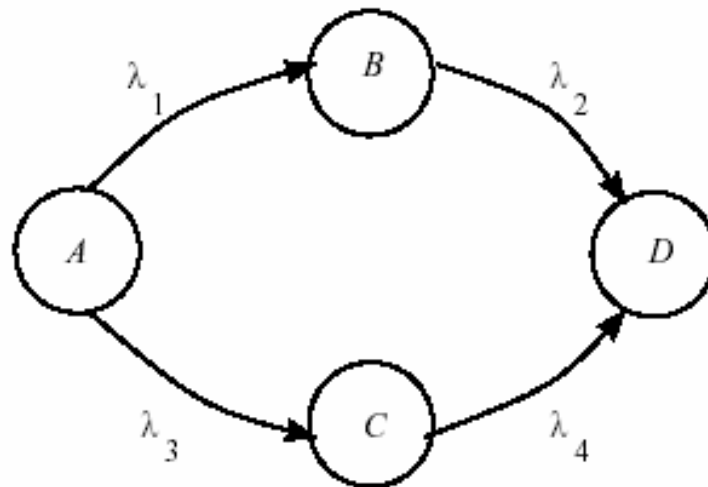
# Graph between $R(t)$ and time



# Operational Security

# Operational Security

- Assuming that at each newly visited node of the privilege graph, the attacker chooses one of the elementary attacks that can be issued from that node only (memoryless property) and assigning to each arc a rate at which the attacker succeeds with the corresponding elementary attack, the privilege graph is transformed into a CTMC.





## Operational Security (Contd.)

- The matrix  $\hat{Q}_R$  obtained from generator matrix  $Q$  by restricting only to the transient states is

$$\hat{Q} = \begin{bmatrix} -(\lambda_1 + \lambda_3) & \lambda_1 & \lambda_3 \\ 0 & -\lambda_2 & 0 \\ 0 & 0 & -\lambda_4 \end{bmatrix}$$

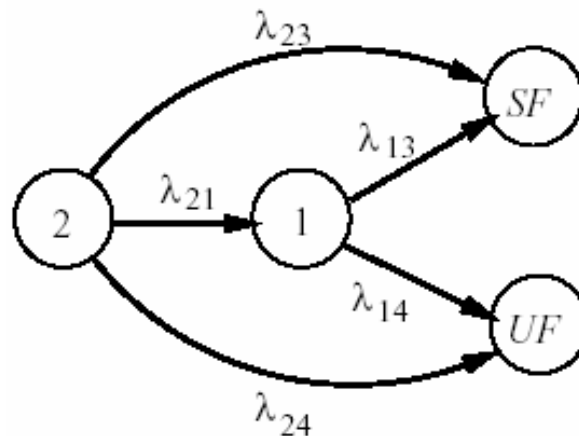
- From this it follows that METF (Mean Effort To Failure) becomes

$$\text{METF} = \sum_{i \in \{A, B, C\}} \tau_i = \frac{1}{\lambda_1 + \lambda_3} \left( 1 + \frac{\lambda_1}{\lambda_2} + \frac{\lambda_3}{\lambda_4} \right).$$

# Recovery Block Architecture

# Recovery Block Architecture

- Consider a recovery block (RB) architecture implemented on a dual processor system that is able to tolerate one hardware fault and one software fault.
- The hardware faults can be tolerated due to the hot standby hardware component with a duplication of the RB software and a concurrent comparator for acceptance tests.



## Recovery Block Architecture (Contd.)

- The transition rates and their meanings are given in the table

Transition rate	Value	Meaning
$\lambda_{21}$	$2c\lambda_H$	covered hardware component failure
$\lambda_{23}$	$2\bar{c}\lambda_H + \lambda_{SD}$	Not covered hardware component failure or detected RB failure
$\lambda_{24}$	$\lambda_{SU}$	undetected RB failure
$\lambda_{13}$	$c\lambda_H + \lambda_{SD}$	detected RB failure or covered hardware component failure
$\lambda_{14}$	$\bar{c}\lambda_H + \lambda_{SU}$	Not covered hardware component failure or undetected RB failure

## Recovery Block Architecture (Contd.)

- The system of differential equation is given by

$$\begin{aligned}\frac{d\pi_2(t)}{dt} &= -(\lambda_{21} + \lambda_{23} + \lambda_{24})\pi_2(t), \\ \frac{d\pi_1(t)}{dt} &= -(\lambda_{13} + \lambda_{14})\pi_1(t) + \lambda_{21}\pi_2(t), \\ \frac{d\pi_{SF}(t)}{dt} &= \lambda_{23}\pi_2(t) + \lambda_{13}\pi_1(t), \\ \frac{d\pi_{UF}(t)}{dt} &= \lambda_{24}\pi_2(t) + \lambda_{14}\pi_1(t),\end{aligned}$$

- Thus reliability of system becomes

$$\begin{aligned}R(t) &= \pi_2(t) + \pi_1(t) \\ &= 2ce^{-(\lambda_H + \lambda_S)t} - (2c - 1)e^{-(2\lambda_H + \lambda_S)t}\end{aligned}$$

$$\text{where } \lambda_S = \lambda_{SD} + \lambda_{SU}$$

## Recovery Block Architecture (Contd.)

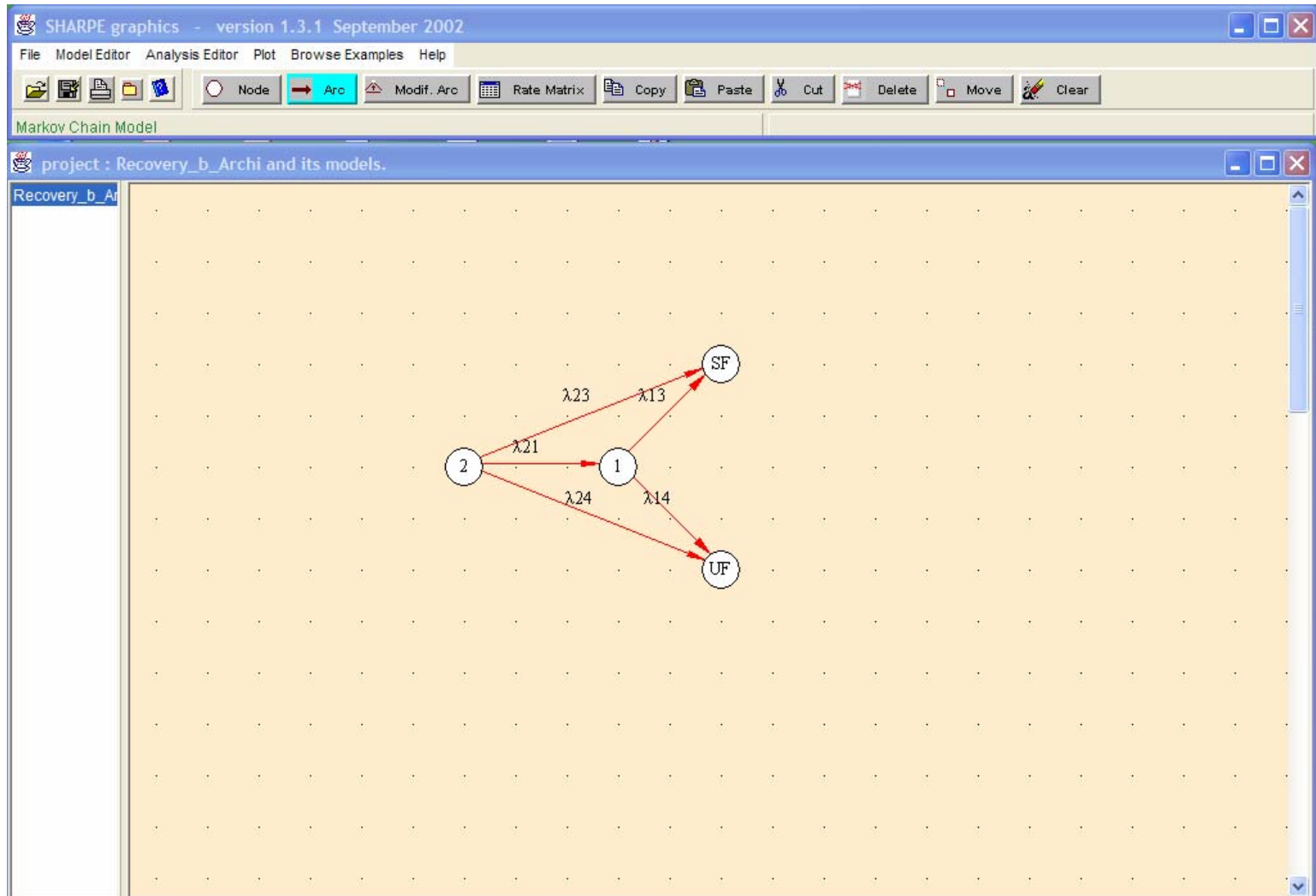
- Similarly, the absorption probability to the safe failure state is:

$$\begin{aligned} P_{SF} &= \pi_{SF}(\infty) \\ &= \frac{2\bar{c}\lambda_H + \lambda_{SD}}{2\lambda_H + \lambda_S} + \frac{2c\lambda_H(c\lambda_H + \lambda_{SD})}{(2\lambda_H + \lambda_S)(\lambda_H + \lambda_S)} \end{aligned}$$

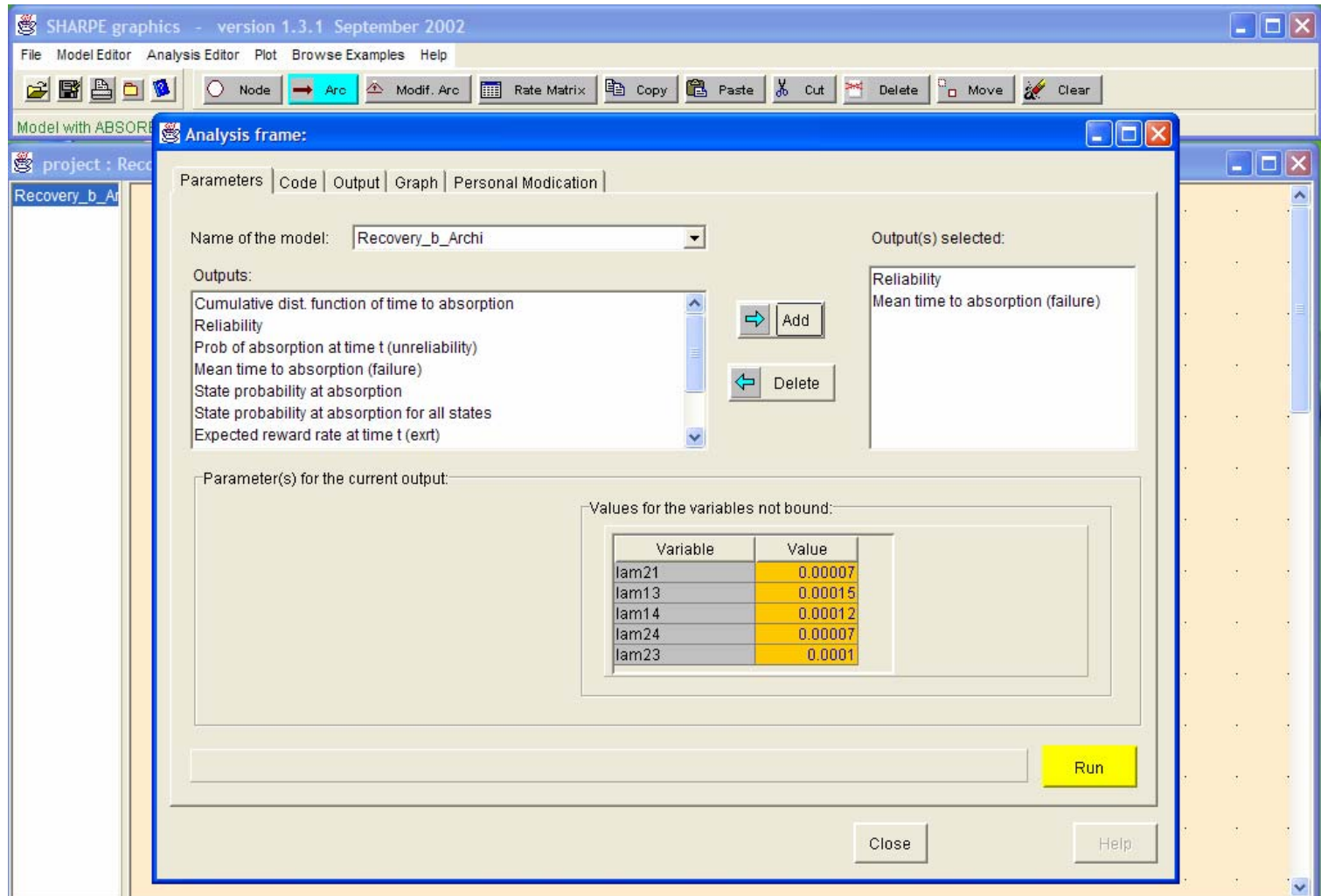
- And the absorption probability to the unsafe failure state is:

$$\begin{aligned} P_{UF} &= \pi_{UF}(\infty) \\ &= \frac{\lambda_{SU}}{2\lambda_H + \lambda_S} + \frac{2c\lambda_H(\bar{c}\lambda_H + \lambda_{SU})}{(2\lambda_H + \lambda_S)(\lambda_H + \lambda_S)} \end{aligned}$$

# Model made in SHARPE GUI



# Parameter assigned and Output asked





# SHARPE Input file

```

• format 8
• factor on

• markov Recovery_b_Archi(lam21, lam13, lam14, lam24, lam23)
• 2 1 lam21
• 2 UF lam24
• 2 SF lam23
• 1 SF lam13
• 1 UF lam14
• end
• * Initial Probabilities defined:
• 2 init_Recovery_b_Archi_2
• 1 init_Recovery_b_Archi_1
• SF init_Recovery_b_Archi_SF
• UF init_Recovery_b_Archi_UF
• end

• * Initial Probabilities assigned:
• bind
•   init_Recovery_b_Archi_2 0
•   init_Recovery_b_Archi_1 0
•   init_Recovery_b_Archi_SF 0
•   init_Recovery_b_Archi_UF 0
• end

• echo

```

Model defined

```

• *****
• *****
• echo ***** Outputs asked for the model *****
• *****

• * Initial Probability: ini
• bind
•   init_Recovery_b_Archi_UF 0
•   init_Recovery_b_Archi_2 1
•   init_Recovery_b_Archi_1 0
•   init_Recovery_b_Archi_SF 0
• end

• bind lam21 0.00007
• bind lam13 0.00015
• bind lam14 0.00012
• bind lam24 0.00007
• bind lam23 0.0001

• func Reliability(t) 1-tvalue(t;Recovery_b_Archi; lam21, lam13, lam14,
• lam24, lam23)
• loop t,1,1000,100
• expr Reliability(t)
• end

• bind lam21 0.00007
• bind lam13 0.00015
• bind lam14 0.00012
• bind lam24 0.00007
• bind lam23 0.0001

• var MTTAb mean(Recovery_b_Archi, UF; lam21, lam13, lam14, lam24,
• lam23)
• expr MTTAb

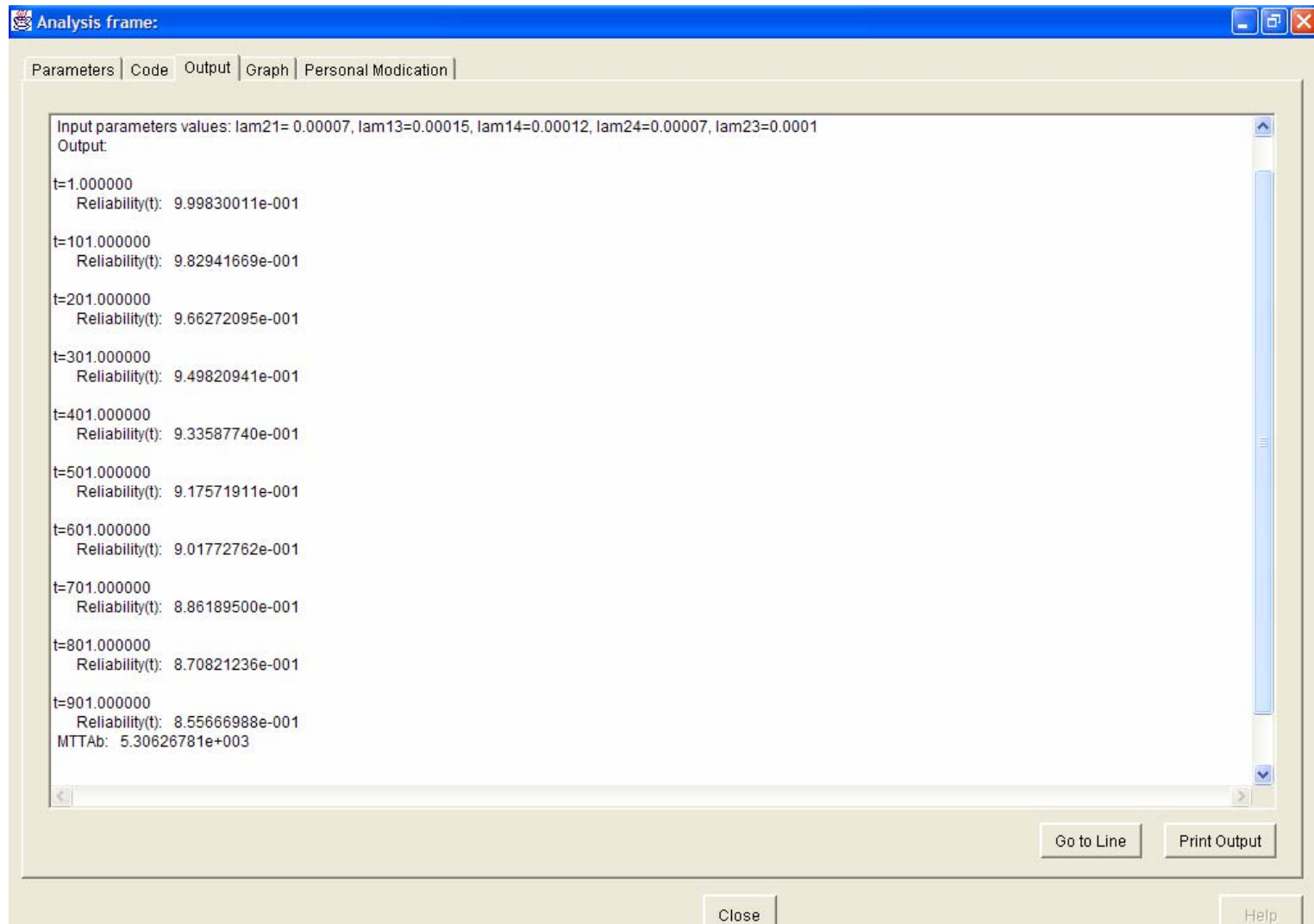
• end

```

Output asked

Initial prob. assigned

# Output generated by SHARPE GUI



The screenshot shows a window titled "Analysis frame:" with a blue title bar. Below the title bar is a menu bar with tabs: "Parameters", "Code", "Output", "Graph", and "Personal Modication". The "Output" tab is selected. The main area contains the following text:

```
Input parameters values: lam21= 0.00007, lam13=0.00015, lam14=0.00012, lam24=0.00007, lam23=0.0001
Output:

t=1.000000
  Reliability(t): 9.99830011e-001

t=101.000000
  Reliability(t): 9.82941669e-001

t=201.000000
  Reliability(t): 9.66272095e-001

t=301.000000
  Reliability(t): 9.49820941e-001

t=401.000000
  Reliability(t): 9.33587740e-001

t=501.000000
  Reliability(t): 9.17571911e-001

t=601.000000
  Reliability(t): 9.01772762e-001

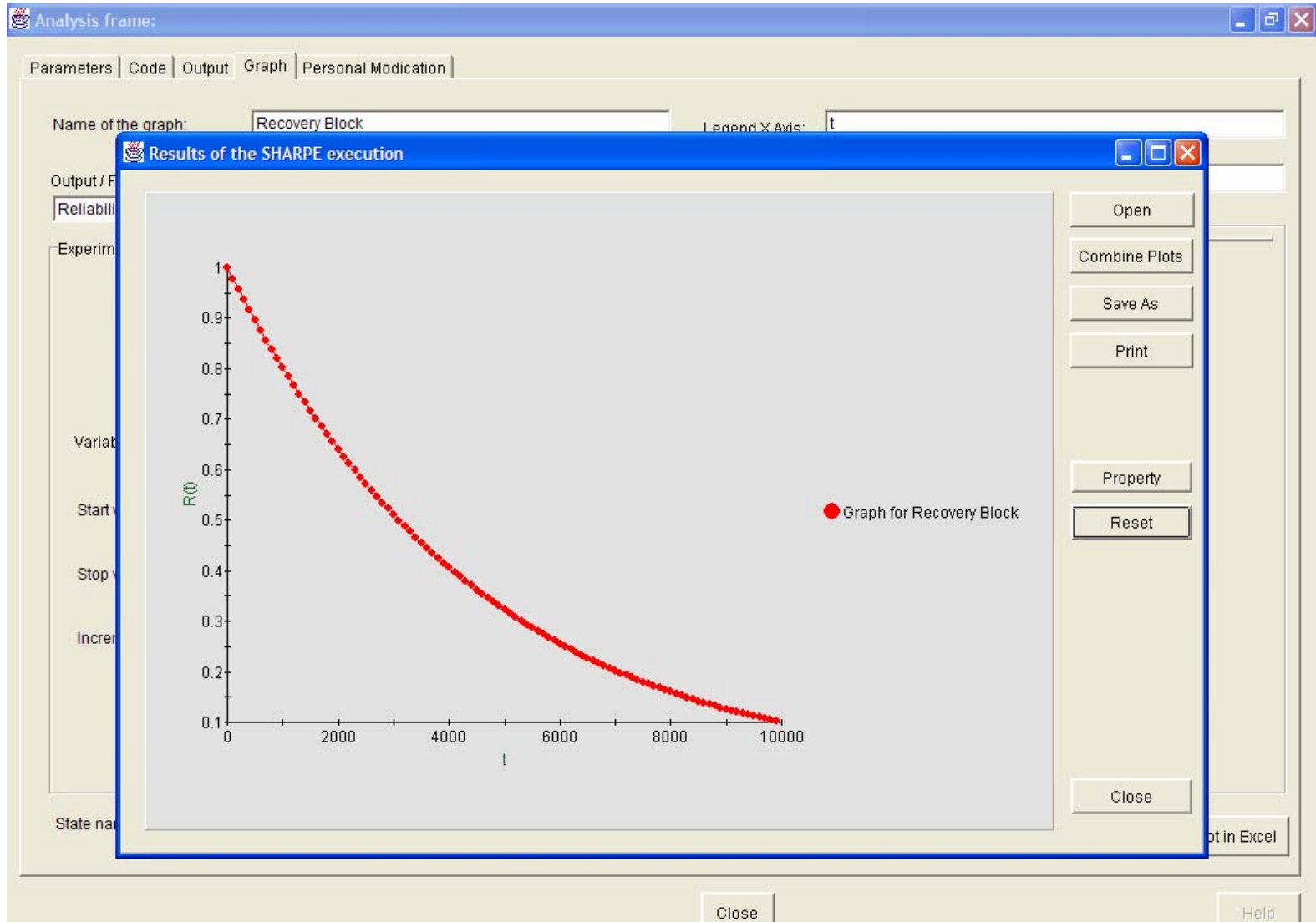
t=701.000000
  Reliability(t): 8.86189500e-001

t=801.000000
  Reliability(t): 8.70821236e-001

t=901.000000
  Reliability(t): 8.55666988e-001
MTTAb: 5.30626781e+003
```

At the bottom of the window, there are two buttons: "Go to Line" and "Print Output". Below the window frame, there are two more buttons: "Close" and "Help".

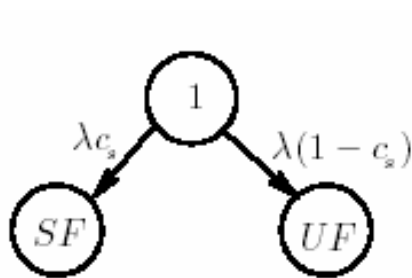
# Plot between $R(t)$ and time



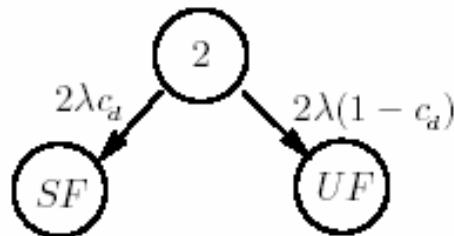
# Conditional MTTF of a Fault-Tolerant System

# Conditional MTTF of a Fault-Tolerant System

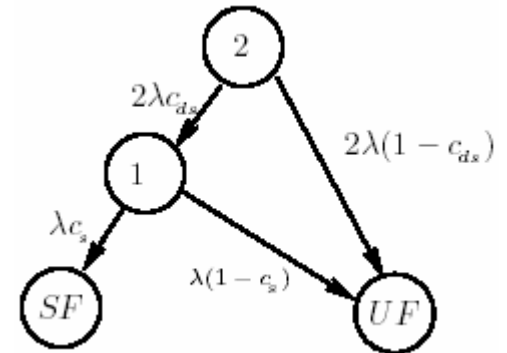
- Consider the homogeneous CTMC models of three commonly used fault-tolerant system architectures.
  - The simplex system  $S$  consists of a single processor.
  - The Duplex system (D) consists of two identical processors executing the same task in parallel.
  - The Duplex system reconfigurable to the simplex system (DS) also consists of two processors executing the same task in parallel.



(a)



(b)



(c)

# Conditional MTTF of a Fault-Tolerant System (Contd.)

- We compare the three architectures with respect to the probability of unsafe failure, the mean time to failure (MTTF) of the system and the **conditional** MTTF to unsafe failure.
- Calculating conditional MTTF  $Q$  matrix becomes

$$Q = \begin{bmatrix} Q_{TT} & Q_{TA} & Q_{TB} \\ \mathbf{0}_{1 \times |T|} & \mathbf{0} & \mathbf{0}_{1 \times |B|} \\ \mathbf{0}_{|B| \times |T|} & \mathbf{0}_{|B| \times 1} & \mathbf{0}_{|B| \times |B|} \end{bmatrix}$$

- Here  $Q_{TT}$  is the partition of the generator matrix consisting of the states in  $T$ ,  $Q_{TA}$  has the transition rates from states in  $T$  to states in  $A$  and similarly  $Q_{TB}$  has the transition rates from states in  $T$  to states in  $B$ .

# Conditional MTTF of a Fault-Tolerant System (Contd.)

- Solving for the three architectures for different parameters we have

<i>Measures</i>	<i>Architecture S</i>	<i>Architecture D</i>	<i>Architecture DS</i>
MTTF	$\frac{1}{\lambda}$	$\frac{1}{2\lambda}$	$\frac{1}{2\lambda} + \frac{c_{ds}}{\lambda}$
$\pi_{UF}(\infty)$	$1 - c_s$	$1 - c_d$	$1 - c_s c_{ds}$
$MTTF_{ UF}$	$\frac{1}{\lambda}$	$\frac{1}{2\lambda}$	$\frac{1 + 2c_{ds} - 3c_s c_{ds}}{2\lambda(1 - c_s c_{ds})}$

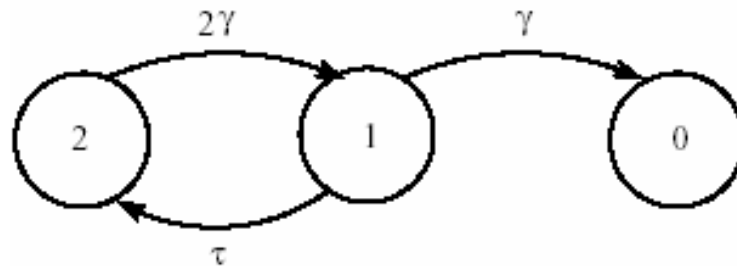
*Dependability measures for the three architectures*

# Real Time System: Multiprocessor Revisited



# Multiprocessor Revisited

- We return to the multiprocessor model earlier discussed but we now consider system failure state '0' as absorbing.



- Since task arrivals occur at the rate  $\lambda$  and task service time is  $EXP(\mu)$ , when the reliability model is in state 2, the performance can be modeled by an  $M/M/2/b$  queue.

## Multiprocessor Revisited (Contd.)

- We make the following reward rate assignment to the states (soft deadline case):

$$r_2 = \lambda[1 - q_b(2)][P(R_b(2) \leq d)],$$

$$r_1 = \lambda[1 - q_b(1)][P(R_b(1) \leq d)]$$

$$r_0 = 0.$$

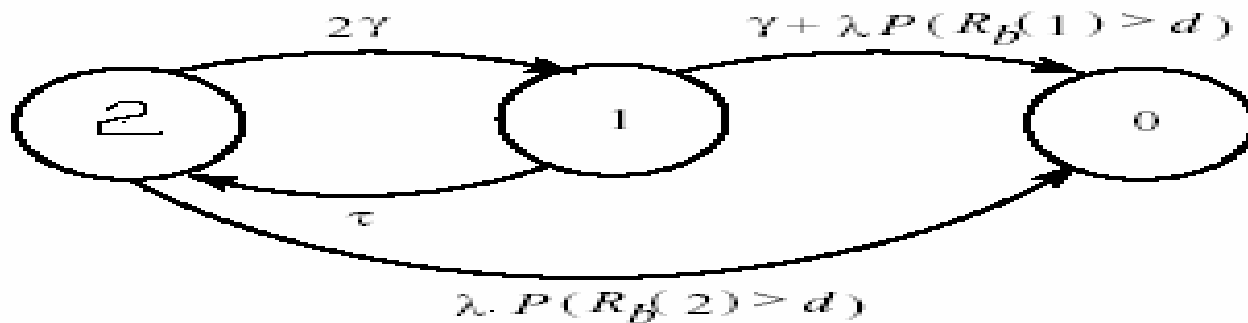
- With this reward assignment, computing the expected accumulated reward until absorption, we can obtain the approximate number of tasks successfully completed until system failure:

$$E[Y(\infty)] = r_2\tau_2 + r_1\tau_1$$

where  $\tau_2$  and  $\tau_1$  are given by equation (8.116) given in the textbook.

## Multiprocessor Revisited (Contd.)

- Now we consider a hard deadline, instead of soft deadline so that if an accepted job fails to complete within the deadline, we will consider the system to have failed.

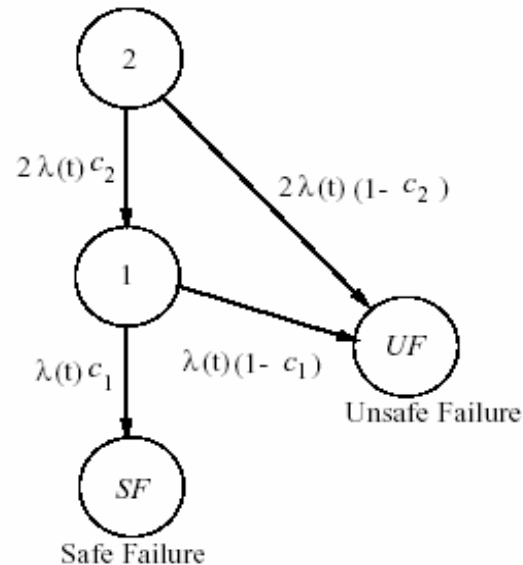


- Note that we have considered the infinite buffer case for simplicity
- Using the  $\tau$  method, we can compute the values of  $\tau_2$  and  $\tau_1$  for the CTMC and the system MTTF that includes the effect of dynamic failures.

# NHCTMC Model of the Duplex System

# NHCTMC Model of the Duplex System

- Consider a *duplex system* with two processors, each of which has a time-dependent failure rate  $\lambda(t) = \lambda_0 \alpha t^{\alpha-1}$ .



- The system shown is a non-homogeneous CTMC, because, as its name suggests, it contains one or more (globally) time-dependent transition rates.

# NHCTMC Model of the Duplex System (Contd.)

- The transient behavior of a NHCTMC satisfies the linear system of first order differential equations:

$$\frac{d\pi(t)}{dt} = \pi(t)Q(t), \quad \text{with } \pi_2(0) = 1.$$

- The Q matrix becomes

$$\begin{aligned} Q(t) &= \begin{bmatrix} -\lambda(t) & 0 & \lambda(t)c_1 & \lambda(t)(1-c_1) \\ 2\lambda(t)c_2 & -2\lambda(t) & 0 & 2\lambda(t)(1-c_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ &= \lambda(t) \begin{bmatrix} -1 & 0 & c_1 & 1-c_1 \\ 2c_2 & -2 & 0 & 2(1-c_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \lambda(t)W, \end{aligned}$$

where

$$W = \begin{bmatrix} -1 & 0 & c_1 & 1-c_1 \\ 2c_2 & -2 & 0 & 2(1-c_2) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

## NHCTMC Model of the Duplex System (Contd.)

- When NHCTMC generator matrix can be factored in this way we can solve the equations simply
- Hence we can define an average failure rate:

$$\bar{\lambda} = \frac{1}{t} \int_0^t \lambda(\tau) d\tau,$$

and get the solution to the NHCTMC by solving a homogeneous CTMC with the generator matrix:

$$\bar{Q} = W\bar{\lambda}.$$

# Software Reliability Growth Models

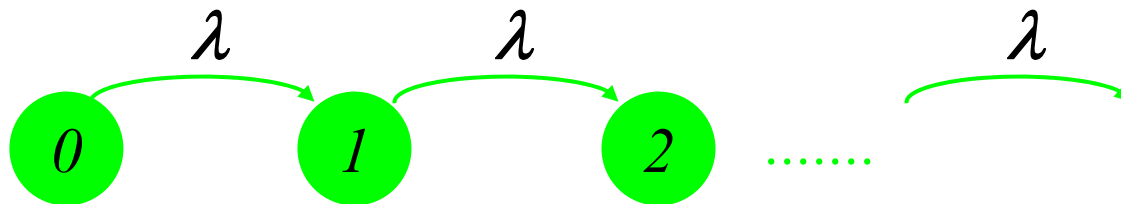


# Software Reliability Growth Models

- Failure data is collected during testing
- Calibrate a reliability growth model using failure data; this model is then used for prediction
- Many SRGMs exist
  - NHPP
  - Jelinski Moranda
- We revisit the above models which we studied in Chapter 5, studying them now as examples of CTMCs.

# Poisson Process

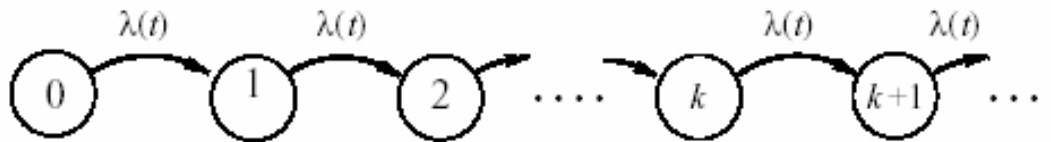
- The Poisson process,  $\{N(t) \mid t \geq 0\}$ , is a homogeneous CTMC (pure birth type) with state diagram shown below
- Since failure intensity is time independent, it cannot capture reliability growth. Hence we resort NHPP.



# Example –Software Reliability Growth Model (NHPP)

- Consider a Nonhomogenous Poisson process (NHPP) proposed by Goel and Okumoto, as a model of software reliability growth during the testing phase. Note that the Markov property is satisfied and it is an example of a non-homogeneous CTMC
- Assume that the number of failures  $N(t)$  occurring in time interval  $(0, t]$  has a time-dependent failure intensity  $\lambda(t)$ .
- Expected number of software failures experienced (and equated to the number of faults found and fixed) by time  $t$ :

$$m(t) = E[N(t)] = \int_0^t \lambda(x) dx$$



# Software Reliability Growth Model

## Finite failure NHPP models

- Finite expected number of faults detected,  $a$ , in an infinite interval
- Expected number of faults detected by time  $t$ , or mean value function, denoted by  $m(t) = ap = aF(t)$
- Failure intensity of the software, denoted by  $\lambda(t)$ :  $\lambda(t) = \frac{dm(t)}{dt}$

- Failure intensity function can also be written as

$$\lambda(t) = af(t) = [a - m(t)]h(t)$$

- $h(t) \rightarrow$  failure occurrence rate per fault (hazard function)
  - $[a - m(t)] \rightarrow$  expected number of faults remaining, non-increasing function of time
- Nature of failure intensity depends on the nature of failure occurrence rate per fault

# Example –Software Reliability Growth Model (NHPP) (Contd.)

- Using previous equation the instantaneous failure intensity can be rewritten by

$$\lambda(t) = af(t) = [a - m(t)]h(t)$$

- This implies that failure intensity is proportional to expected no. of undetected faults at 't'
- Many commonly used NHPP software reliability growth models are obtained by choosing different failure intensities  $\lambda(t)$ , e.g. Goel-Okumoto, Musa-Okumoto model etc.

# Software Reliability Growth Model

## *Finite failure NHPP models*

- Nature of the failure occurrence rate per fault and the corresponding NHPP model

– Constant :

- Goel-Okumoto model

$$h(t) = b$$

– Increasing :

- S-shaped model

$$h(t) = \frac{g^2 t}{1 + gt}$$

- Generalized Goel-Okumoto model

$$h(t) = bct^{c-1}, \quad c > 1$$

– Decreasing :

- Generalized Goel-Okumoto model

$$h(t) = bct^{c-1}, \quad c < 1$$

– **Increasing/Decreasing :**

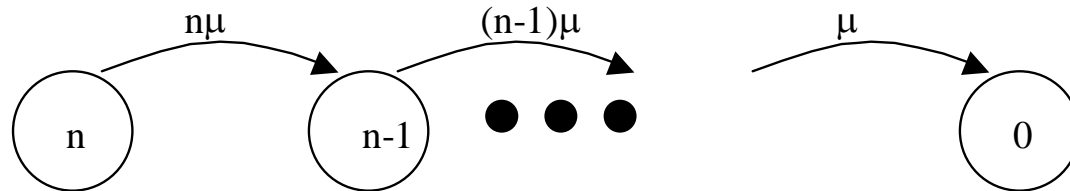
- **Log-logistic model**

$$h(t) = \frac{\lambda \kappa (\lambda t)^{\kappa-1}}{1 + (\lambda t)^\kappa}, \quad \kappa > 1$$

# Example- Jelinski Moranda Model

- This model is based on the following assumptions:
  - The number of faults introduced initially into the software is fixed, say,  $n$ .
  - At each failure occurrence, the underlying fault is removed immediately and no new faults are introduced.
  - Failure rate is state-dependent and is proportional to the number of remaining faults, that is,  $\mu_i = i\mu$ ,  $i = 1, 2, \dots, n$ .
- Model can be described by pure death process
- The constant of proportionality  $\mu$  denotes the failure intensity contributed by each fault, which means that all the remaining faults contribute the same amount to the failure intensity.

# Example- Jelinski Moranda Model (Contd.)



- The mean-value function is given by

$$m(t) = \sum_{k=0}^n k\pi_{n-k}(t) = n(1 - e^{-\mu t})$$

- This can be seen as the expected reward rate at time  $t$  after assigning reward rate  $r_i = n-i$  to state  $i$ .