Fault Trees

Fault Trees

- FT considers the combination of events that may lead to an unsdesirable situation of the system
  (the delivery of improper service for a Reliability study,
  catastrophic failures for a Safety study)

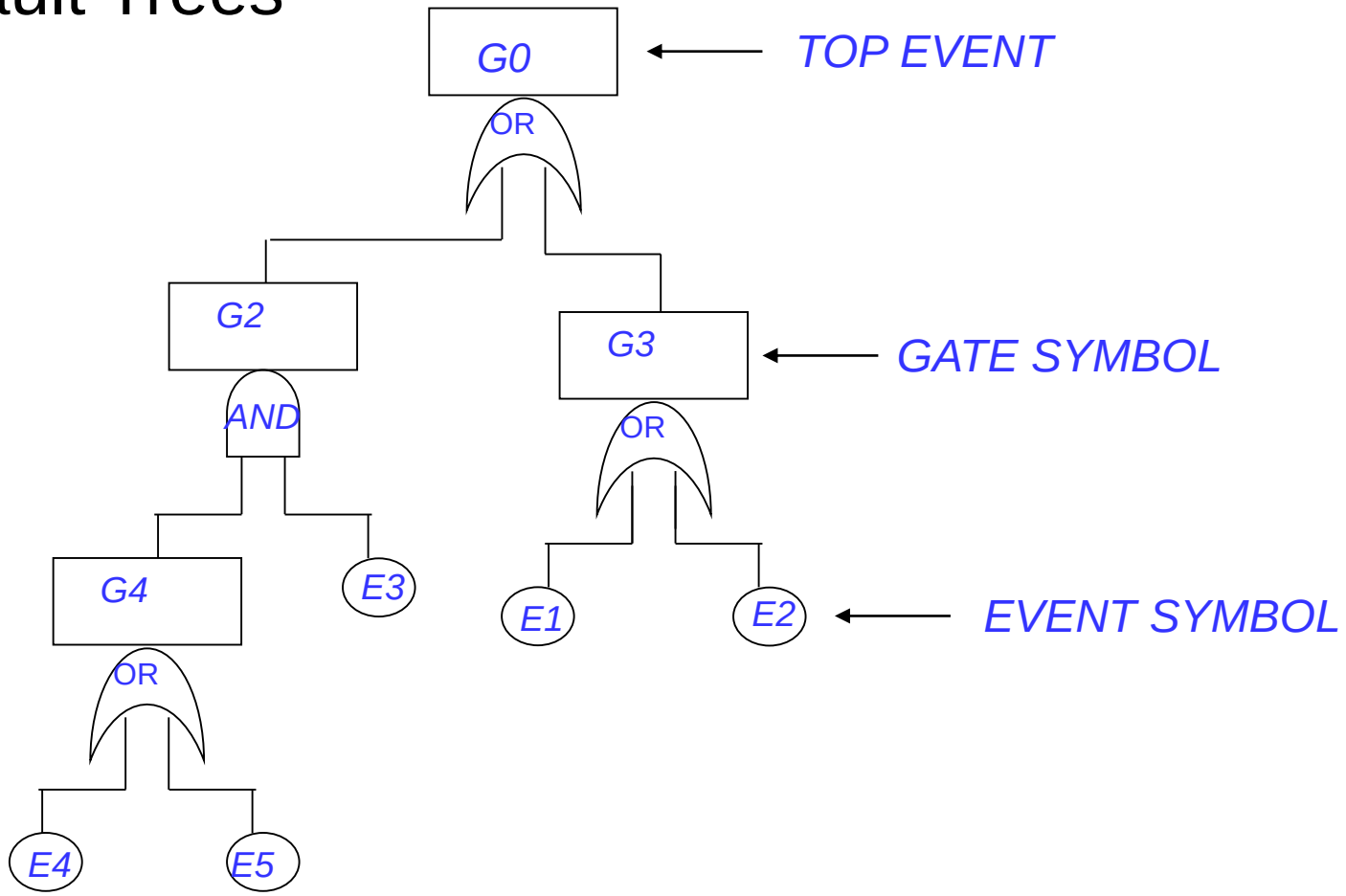
- Describe the scenarios of occurrence of events at abstract level

- Hierarchy of levels of events linked by logical operators

- The analysis of the fault tree evaluates the probability of occurrence of the root event, in terms of the status of the leaves (faulty/non faulty)

- Applicable both at design phase and operational phase

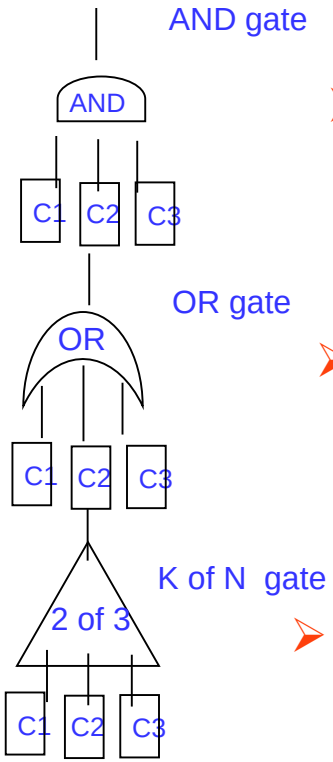
Fault Trees



Describes the Top Event (status of the system) in terms of the status (faulty/non faulty) of the Basic events (system's components)

Fault Trees

- Components are leaves in the tree
- Component faulty corresponds to logical value **true**, otherwise **false**
- Nodes in the tree are boolen AND, OR and k of N gates
- The system fails if the root is true

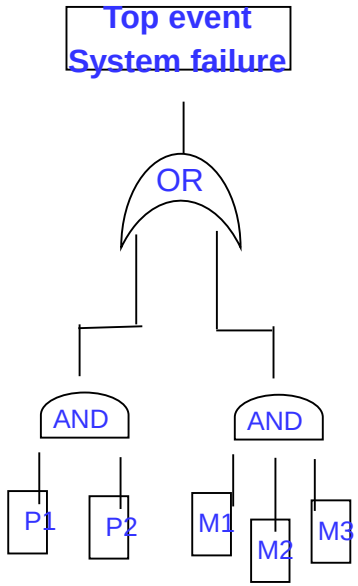


- True if all the components are true (faulty)
- True if at least one of the components is true (faulty)
- True if at least k of the components are true (two or three components) (faulty)

Example:

Multiprocessor with 2 processors and three shared memories

-> the computer fail if all the memories fail or all the processors fail



A cut is defined as a set of elementary events that, according to the logic expressed by the FT, leads to the occurrence of the root event.

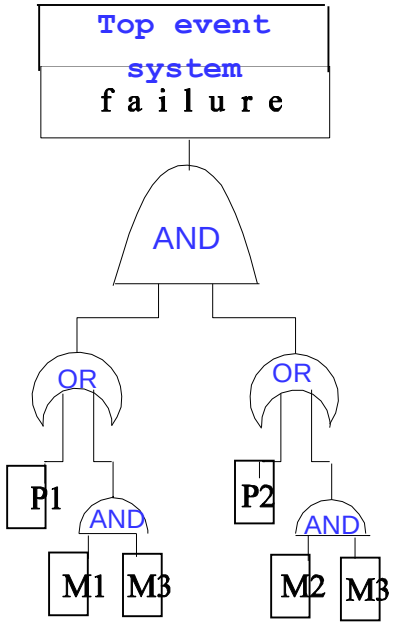
To estimate the probability of the root event, compute the probability of occurrence for each of the cuts and combine these probabilities

Conditioning Fault Trees

- If the same component appears more than once in a fault tree, it violates the independent failure assumption (conditioned fault tree)

- Example

  Multiprocessor with 2 processors and three memories: M1 private memory of P1 M2 private memory of P2, M3 shared memory.



- Assume every process has its own private memory plus a shared memory.
- Operational condition: at least one processor is active and can access to its private or shared memory.
- **repeat** instruction:given a component C whether or not the component is input to more than one gate, the component is unique
  M3 is a shared memory

Conditioning Fault Trees

If a component C appears multiple times in the FT

$$Q_s(t) = Q_{S|C\ Fails}(t)\ Q_C(t) + Q_{S|C\ not\ Fails}(t)\ (1-Q_C(t))$$

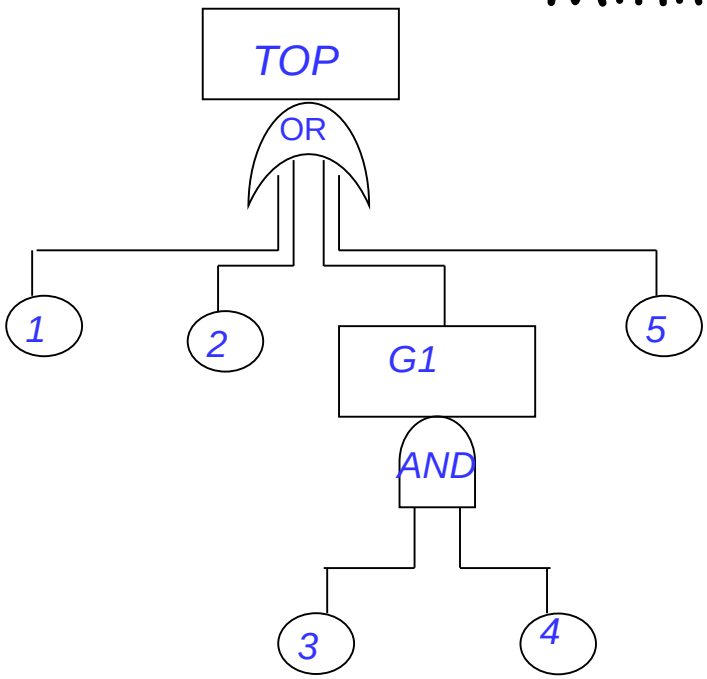
where

S|C Fails is the system given that C fails

and

S|C not Fails is the system given that C has not failed

Minimal cut sets



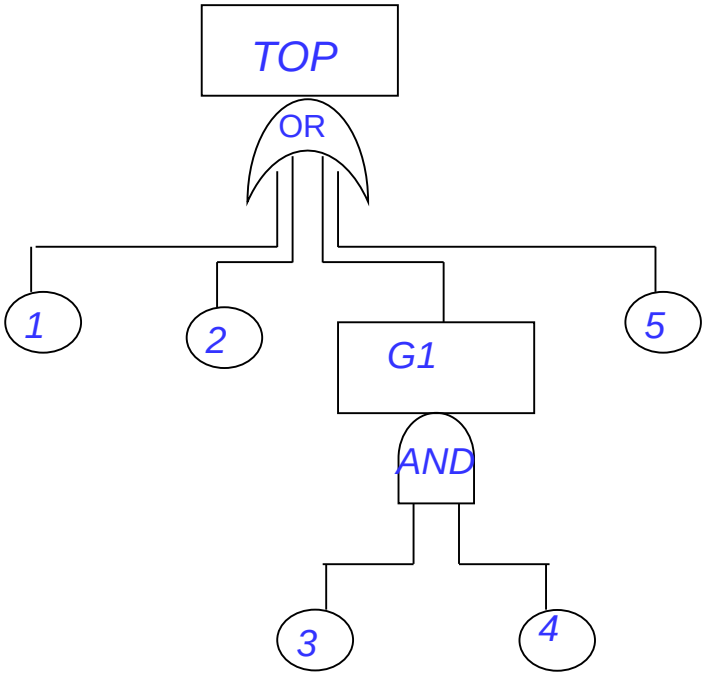
A cut is defined as a set of elementary events that, according to the logic expressed by the FT, leads to the occurrence of the root event.

Cut Sets

Top = {1}, {2} , {G1} , {5} = {1}, {2} , {3, 4} , {5}

Minimal Cut Sets

Top = {1}, {2} , {3, 4} , {5}



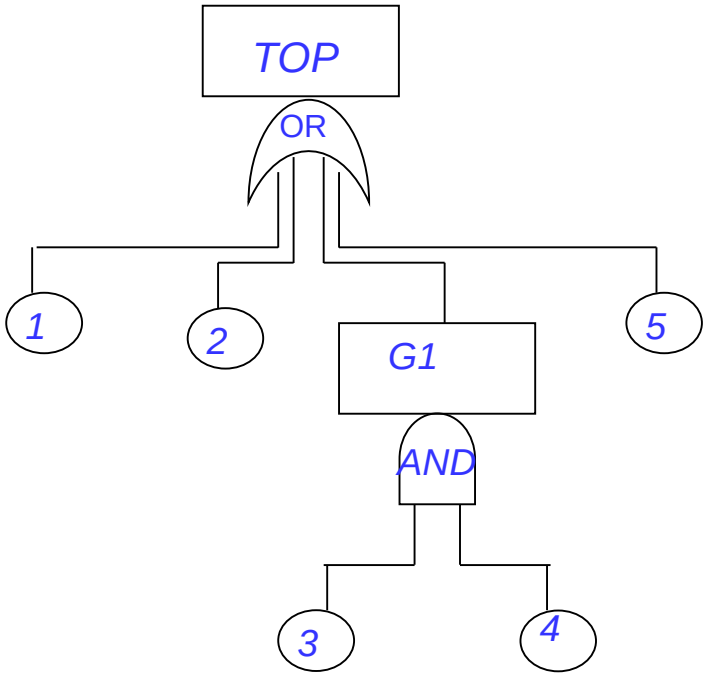
Minimal Cut Sets
Top = {1}, {2} , {3, 4} , {5}

independent faults of the components

$Q_{Si}(t)$ = probability that all components in the minimal cut set *Si* are faulty

$Q_{Si}(t) = q_1(t)\, q_2(t) \ldots q_{ni}(t)$ with *Si* ={1, 2, …, ni }

The numerical solution of the FT is performed by computing the probability of occurrence for each of the cuts, and by combining those probabilities to estimate the probability of the root event



Minimal Cut Sets

Top = {1}, {2} , {3, 4} , {5}

$S_1 = \{1\}$

$S_2 = \{2\}$

$S_3 = \{3, 4\}$

$S_4 = \{5\}$

$$Q_{Top}(t) = Q_{S1}(t) + \ldots + Q_{Sn}(t)$$

n number of mininal cut sets

Fault Trees

- Definition of the Top event
- Analysis of failure models of components
- Minimal cut set

  minimal set of events that leads to the top event

  -> critical path of the system

Analysis:

- Failure probability of Basic events
- Failure probability of minimal cut sets
- Failure probability of Top event
- Single point of failure of the system: minimal cuts with a single event

Model-based evaluation of dependability

State-based models: Markov models

Characterize the state of the system at time t:

- identification of system states
- identification of transitions that govern the changes of state within a system

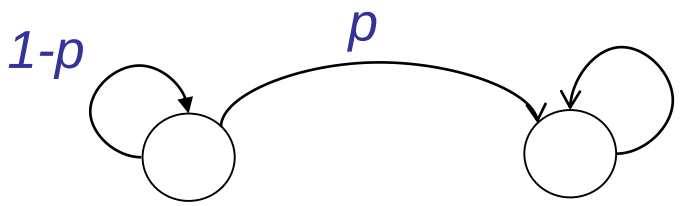
Each state represents a distinct combination of failed and working modules

The system goes from state to state as modules fail and repair.

The state transitions are characterized by the probability of failure and the probability of repair

Markov model:

graph where nodes are all the possible states and arcs are the possible transitions between states (labeled with a probability function)



Reliability model

Each state represents a distinct combination of working and failed components

As time passes, the system goes from state to state as modules fails and are repaired

Allow Reliability/Availability modelling

Model-based evaluation of dependability

Markov models (a special type of random process) :

Basic assumption: the system behavior at any time instant depends only on the current state (independent of past values)

Main points:

- systems with arbitrary structures and complex dependencies can be modeled

- assumption of independent failures no longer necessary

- can be used for both reliability and availability modeling

Random process

In a general random process $\{X_t\}$, the value of the random variable X_{t+1} may depend on the values of the previous random variables $X_{t0}\, X_{t1}$ .....X_t.

Markov process

the state of a process at time t+1 depends only on the state at time t, and is independent *on any state before t.*

$$\mathcal{P}\{X_{t+1} = j | X_0 = k_0, ..., X_{t-1} = k_{t-1}, X_t = i\} = \mathcal{P}\{X_{t+1} = j | X_t = i\}$$

Markov property: "the current state is enough to determine the future state"

Markov chain

A **Markov chain** is a Markov process X with *discrete state space S*.

A Markov chain is homogeneous if it has *steady-state transition probabilities*:

$$\mathcal{P}\{X_{t+1} = j | X_t = i\} = \mathcal{P}\{X_1 = j | X_0 = i\} \ \forall t \geq 0$$

The probability of transition from state i to state j does not depend by the time. This probability is called p_{ij}

$$p_{ij} = \mathcal{P}\{X_1 = j | X_0 = i\}$$

We consider only *homogeneous* Markov chains

Discrete-time Markov chains (DTMC)

Continuous-time Markov chains (CTMC)

Discrete-time Markov model of a simplex system with repair

$\{X_t\}$ t=0, 1, 2, …. S={0, 1}

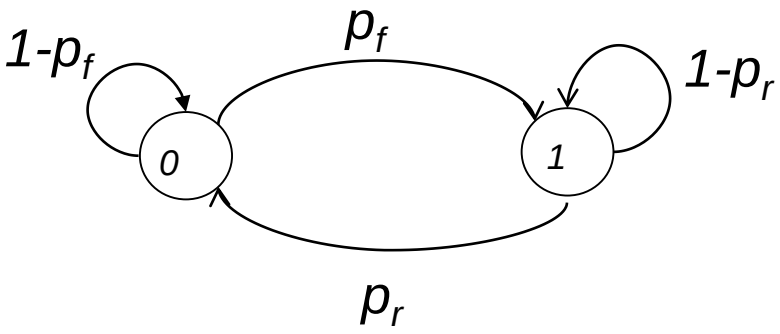
State 0 : working
State 1: failed

- all state transitions occur at fixed intervals
- probabilities assigned to each transition

p_f *Failure probability*

p_r *Repair probability*

The probability of state transition depends only on the current state



Graph model

$$P = \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} 1-p_f & p_f \\ p_r & 1-p_r \end{bmatrix} \end{array}$$

new state

current state

Transition Probability Matrix

- Pij = probability of a transition from state i to state j
- Pij >=0
- the sum of each row must be one

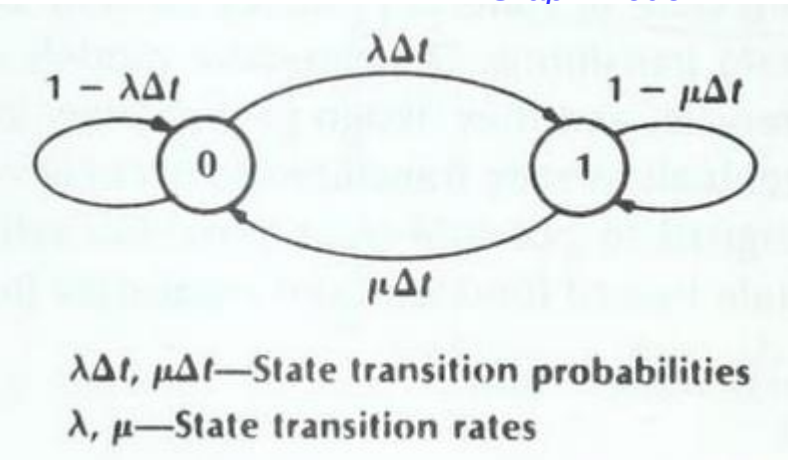
Continuous-time Markov model of a simplex system with repair

derived from the discrete time model, taking the limit as the time-step interval approaches zero

λ failure rate, μ repair rate

state 0: working
state 1: failed

Graph model



$$P = \begin{bmatrix} 1 - \lambda\Delta t & \lambda\Delta t \\ \mu\Delta t & 1 - \mu\Delta t \end{bmatrix}$$

Transition Matrix P

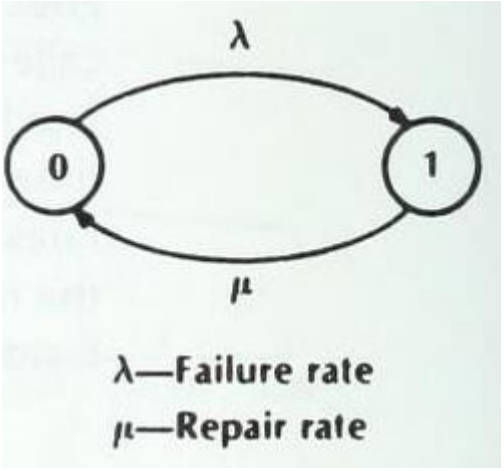
Continuous-time Markov models

Matrix form:

T matrix

$$[\dot{p}_0(t), \dot{p}_1(t)] = [p_0(t), p_1(t)] \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

The set of equations can be written by inspection of a transition diagram without self-loops and Δt's:



Continuous time Markov model graph

The change in state 0 is minus the flow out of state 0 times the probability of being in state 0 at time t, plus the flow into state 0 from state 1 times the probability of being in state 1.

Continuous-time Markov models

$$p_0(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}$$

$$p_1(t) = \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t}$$

A(t)

$p_0(t)$ probability that the system is in the operational state at time t, availability at time t

The availability consists of a steady-state term and an exponential decaying transient term

Steady-state solution

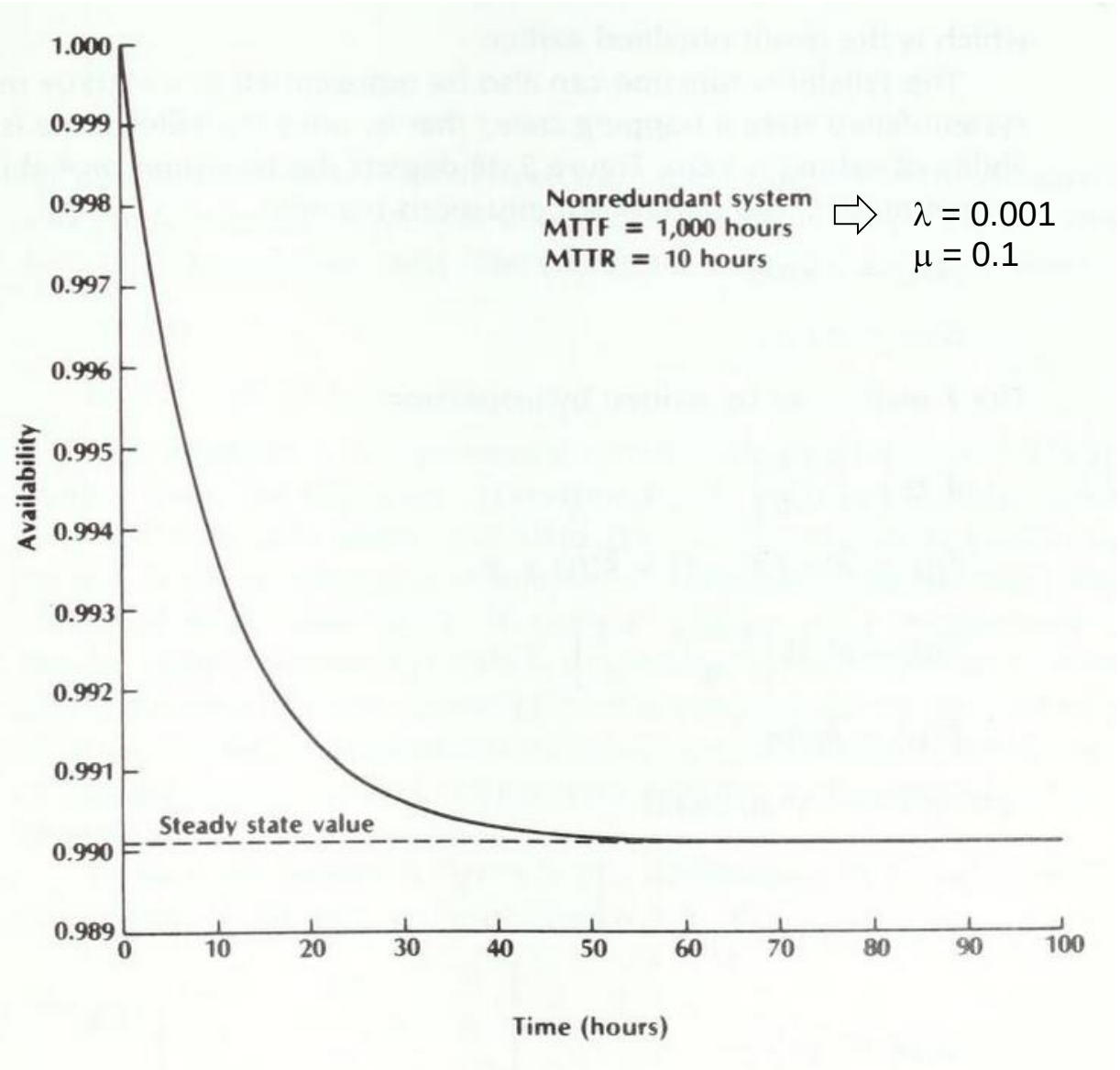
Chapman-Kolmogorov equations: derivative replaced by 0; p0(t) replaced by p0(0) and p1(t) replaced by p1(0)

$$0 = -\lambda p_0 + \mu p_1$$

$$0 = \lambda p_0 - \mu p_1$$

$$p_0 = \frac{1}{1 + \frac{\lambda}{\mu}} = \frac{\mu}{\lambda + \mu}$$

Availability as a function of time

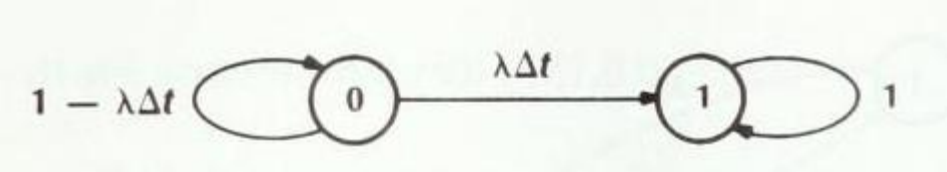


The steady-state value is reached in a very short time

Continuous-time Markov models: Reliability

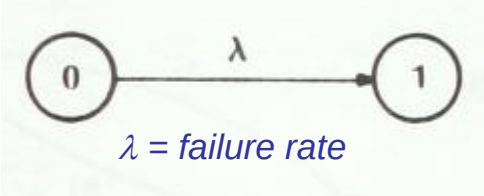
Markov model making the system-failed state a trapping state

Single system without repair



$\lambda\Delta t$ = state transition probability

Continuous time Markov model graph



λ = failure rate

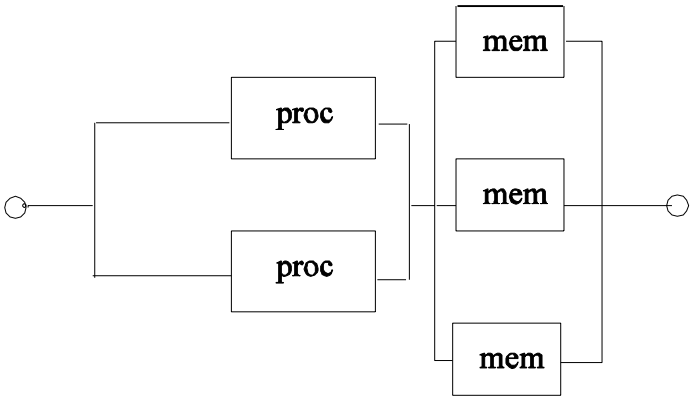
T matrix

$$\begin{bmatrix} -\lambda & \lambda \\ 0 & 0 \end{bmatrix}$$

T matrix can be built by inspection

An example of modeling (CTMC)

Multiprocessor system with 2 processors and 3 shared memories system. System is operational if at least one processor and one memory are operational.



λ_m failure rate for memory

λ_p failure rate for processor

X random process that represents the number of operational memories and the number of operational processors at time t

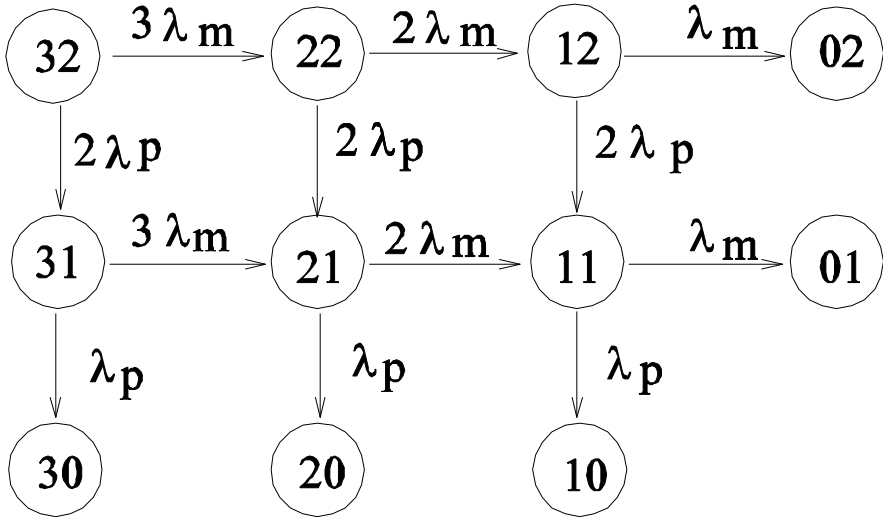
Given a state (i, j):

i is the number of operational memories;

j is the number of operational processors

S = {(3,2), (3,1), (3,0), (2,2), (2,1), (2,0), (1,2), (1,1), (1,0), (0,2), (0,1)}

Reliability modeling



λ_m failure rate for memory
λ_p failure rate for processor

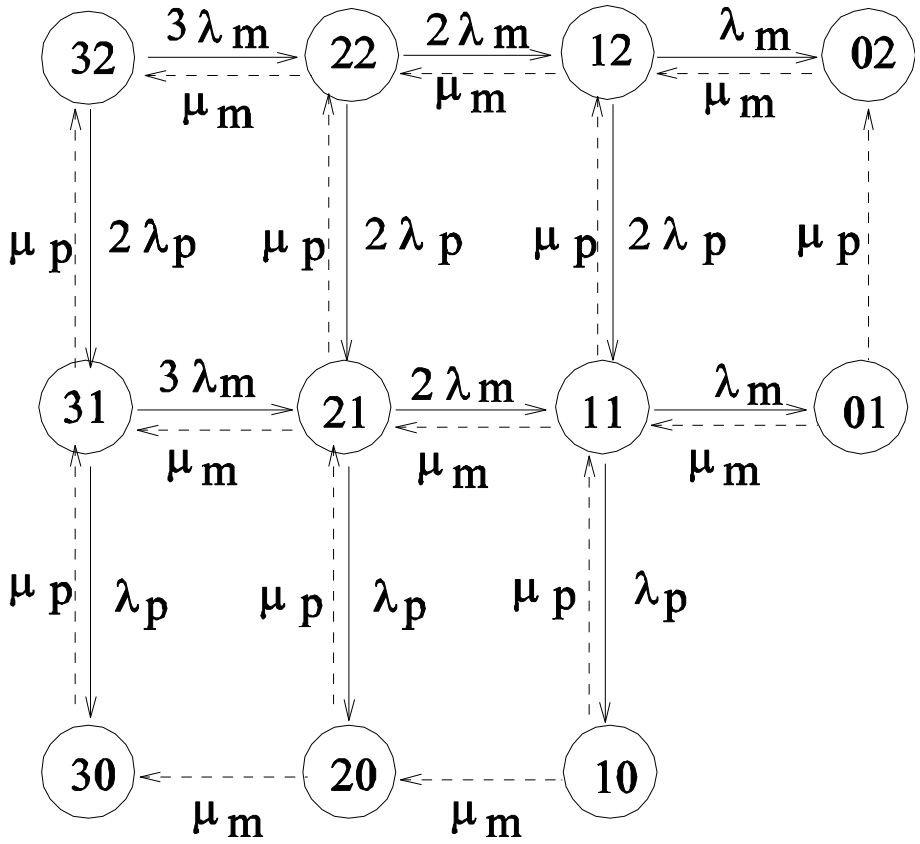
(3, 2) -> (2,2) failure of one memory

(3,0), (2,0), (1,0), (0,2), (0,1) are absorbent states

Availability modeling

- Assume that faulty components are replaced and we evaluate the probability that the system is operational at time t
- Constant repair rate μ (number of expected repairs in a unit of time)
- Strategy of repair:
  only one processor or one memory at a time can be substituted
- The behaviour of components (with respect of being operational or failed) is not independent: it depends on whether or not other components are in a failure state.

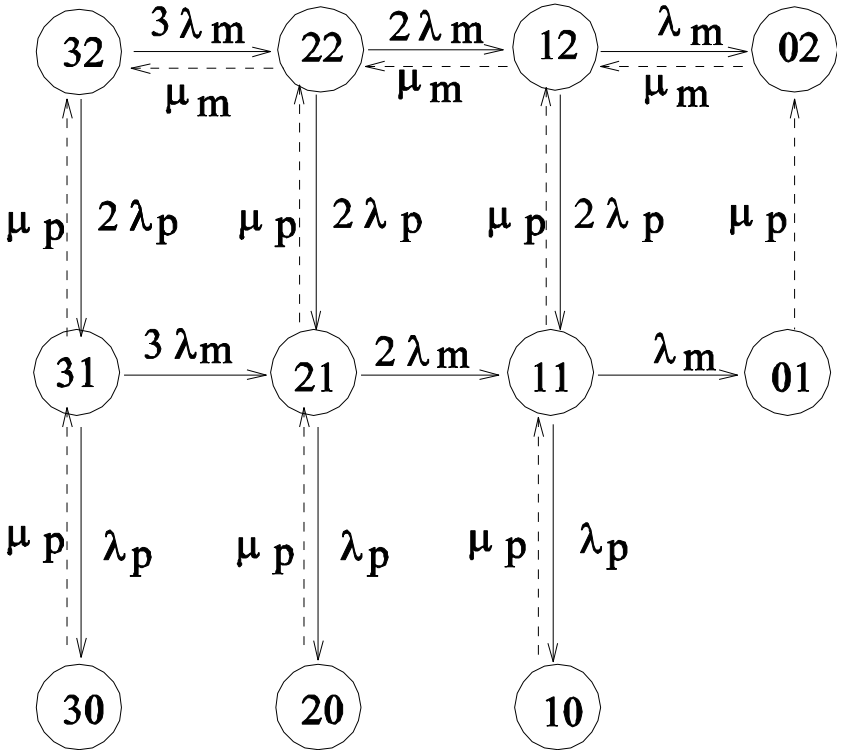
- Strategy of repair:
  only one component can be substituted at a time



λm failure rate for memory
λp failure rate for processor
μm repair rate for memory
μp repair rate for processor

- An alternative strategy of repair:

  only one component can be substituted at a time and processors have higher priority
- exclude the lines μm representing memory repair in the case where there has been a process failure



State occupacy vector

$\pi^{(t)} = [\pi_0^{(t)}, \pi_1^{(t)}, \pi_2^{(t)}, \ldots]$ state occupancy vector

Transient analysis

$\pi_j^{(t)}$ prob of being in state j at time t

Steady-state behaviour

$$\lim_{t \to \infty} \pi_j^{(t)}$$

Many solution methods exist

Moebius tool

Instructions to obtain the Moebius tool:
1. visit http://www.mobius.illinois.edu/ .
2. Click "Login" in the menu bar.
3. In the login page, click "Create an account".
4. Follow instructions to obtain a license. IN PARTICULAR:
a) use an unipi.it email address if possible (not commercial addresses like gmail);
b) in a comment field, say that you attend a course on the Mobius tool held by Cinzia Bernardeschi (owner of an academic licence).
5. Within 48 hours, you should receive a confirmation letter with the link to download the tool.
6. Versions are available for Ubuntu Linux, Mac OSX, and Windows, either 32 or 64 bit.