ECE 454/544: Fault-Tolerant Computing & Reliability Engineering

Lecture #16–

Markov-based Safety Analysis & Problems and Solution

Instructor: Dr. Liudong Xing

Fall 2022

Administrative Issues
(Nov. 16, Wednesday)

- Homework#7 (last one☺)
  - Due by **Nov. 21, Monday**
- Project final report
  - Due by **Nov. 30, Wednesday**
  - Please check out the Report Guidelines for requirements

Review of Lecture #15

- A Markov process is a stochastic process with Markov property: probabilities of future states depend only on the current state and not on the history
- Any fault tree model (static or dynamic) with exponential component failure distribution can be solved as a Markov chain (with four steps)
  - Step 1: convert the fault tree model to a Markov chain
  - Step 2: find the state equations of the Markov chain
  - Step 3: find state probabilities by solving the state equations
    - Asymptotic (steady-state/long-run) solution
    - Time-dependent solution
  - Step 4: find the system reliability or unreliability

Agenda

- **Markov-based safety analysis**
- Markov analysis
  - Pros & Cons
  - Solutions to address the Cons

Safety Concepts (L#1 revisit)

- Safety, *S(t)* -- the probability that a system *either* performs correctly *or* discontinues its operations in a "safe" manner.
  - Not disrupt the operation of other systems
  - Cause no harm to any people associated with the system
- Safety is the probability that a safe action will result after a failure occurs

5

Safety Concepts

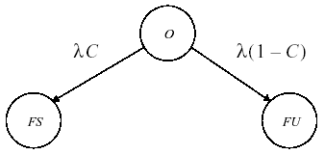
- Concepts of safe and unsafe highly depend on the applications
  - The definition of safe and unsafe failures must be created uniquely for each application
- Fundamental concept of safety analysis is that the system will possess two different ways in which it can fail
  - System fails safely vs. system fails unsafely

6

Safety Modeling and Analysis

- Markov models are usually required to model the safety
  - Splitting the system failed states into two separate states: failed safe (FS) and failed unsafe (FU)

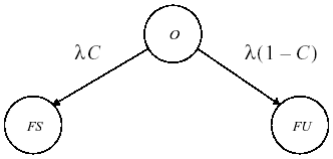
- Example: a simplex system containing a hardware module with a failure rate λ & self-diagnostics with a fault detection coverage of C.
  - Safe failures: failures are detected by the self-diagnostics
  - Unsafe failures: failures are not detected by the self-diagnostics



Markov safety model

Safety Analysis of a Simplex System

- Solution:



Markov safety model

$$\begin{bmatrix} \frac{d}{dt}P_O(t) \\ \frac{d}{dt}P_{FS}(t) \\ \frac{d}{dt}P_{FU}(t) \end{bmatrix} = \begin{bmatrix} -\lambda & 0 & 0 \\ \lambda C & 0 & 0 \\ \lambda(1-C) & 0 & 0 \end{bmatrix} \begin{bmatrix} P_O(t) \\ P_{FS}(t) \\ P_{FU}(t) \end{bmatrix}$$

State equations

- Assume $P_O(0)=1$, $P_{FS}(0)=P_{FU}(0)=0$

Safety Analysis of a Simplex System (Cont'd)

- Taking the Laplace transform of the state equations provides

$$\begin{bmatrix} sP_O(s)-1 \\ sP_{FS}(s) \\ sP_{FU}(s) \end{bmatrix} = \begin{bmatrix} -\lambda P_O(s) \\ \lambda C P_O(s) \\ \lambda(1-C)P_O(s) \end{bmatrix} \Rightarrow \begin{bmatrix} P_O(s) \\ P_{FS}(s) \\ P_{FU}(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{s+\lambda} \\ C\left(\frac{1}{s}-\frac{1}{s+\lambda}\right) \\ (1-C)\left(\frac{1}{s}-\frac{1}{s+\lambda}\right) \end{bmatrix}$$

- Taking the inverse Laplace transform generates

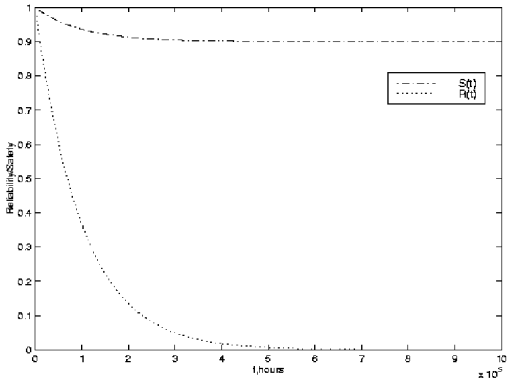
$$\begin{bmatrix} P_O(t) \\ P_{FS}(t) \\ P_{FU}(t) \end{bmatrix} = \begin{bmatrix} e^{-\lambda t} \\ C(1-e^{-\lambda t}) \\ (1-C)(1-e^{-\lambda t}) \end{bmatrix}$$

- Thus,
  - System reliability $R(t) = P_O(t) = e^{-\lambda t}$
  - System safety $S(t) = P_O(t) + P_{FS}(t) = C + (1-C)e^{-\lambda t}$

9

Safety Analysis of a Simplex System (Cont'd)

- Assume λ=1e-5 failures / hour and C=0.9



$R(t) = P_O(t) = e^{-\lambda t}$

$S(t) = P_O(t) + P_{FS}(t) = C + (1-C)e^{-\lambda t}$

- Professional society: The International System Safety Society

http://www.system-safety.org

10

Agenda

✓ Markov-based safety analysis

- Markov analysis
  - **Pros & Cons**
  - Solutions to address the Cons

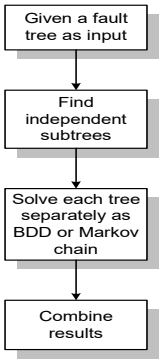
Pros. of Markov Models

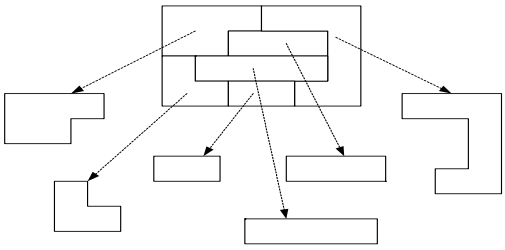
- Powerful in terms of modeling capability, as compared with combinatorial models (RBD, BDD, static fault trees, cut-sets, etc)
  - Not restricted to only two possible states of the component
  - Allow for easily modeling
    - various dependencies (FDEP, HSP, WSP, CSP, PAND)
    - rather complicated repair strategies
    - fault imperfect coverage

Cons of Markov Models

- Limited to exponential time-to-failure distributions
- State explosion problem
  - The number of system states increase exponentially with the size and complexity of the system → intractable models
  - Suitable only for relatively small systems
- **Solutions**
  - Modularization
  - Bounding method

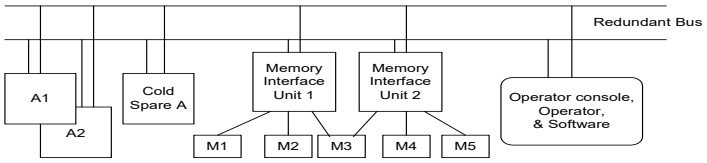
Solution #1: Modular Approach



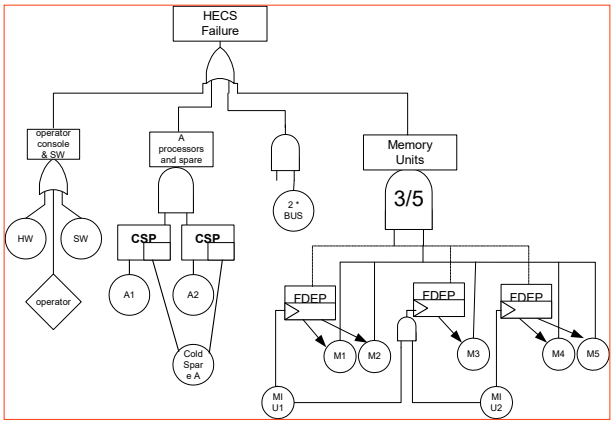


- Modularization combines the best of combinatorial (BDD) and Markov approaches
  - Use fast and efficient BDD approach for static modules
  - Build Markov chain automatically when needed for dynamic behavior
  - divide-and-conquer helps avoid models which are too large to solve

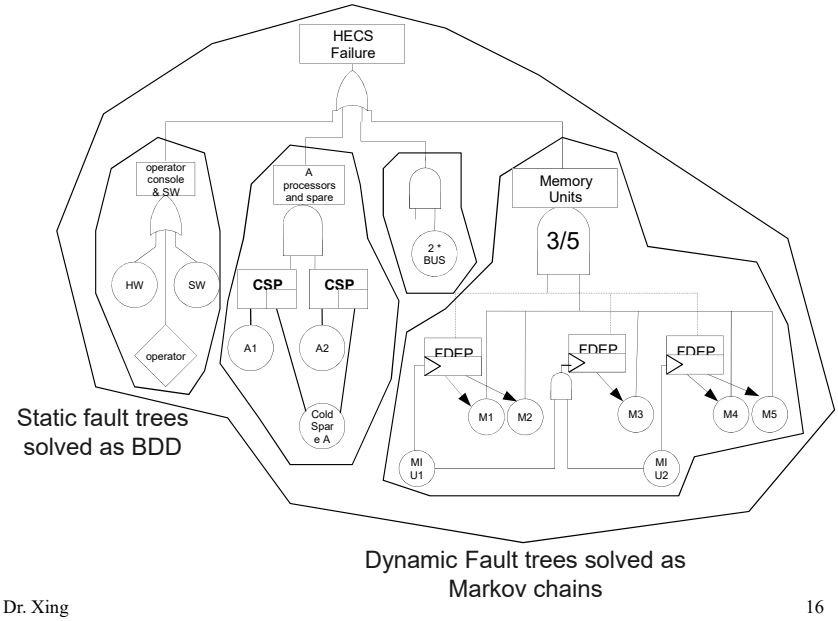
HECS Dynamic Fault Tree (L#15)



- Processors A1 and A2 share the cold spare A
- 3 of the 5 memory units are needed; if MIU fails, memory is not accessible
- At least one bus is required



Modularized HECS Fault Tree



References

Rohit Gulati, *A modular approach to static and dynamic fault tree analysis*, Master's thesis, University of Virginia Department of Electrical Engineering, 1996.

Rohit Gulati and Joanne Bechta Dugan, *A modular approach for analyzing static and dynamic fault trees*, Proceedings of the Reliability and Maintainability Symposium, January, 1997, pp. 57–63.

Solution #2: Bounding Methods

- Reduce the number of states required in the model to a more manageable level
- Only a portion of the state space of Markov chains is generated for solution by employing some state truncation techniques, for example,
  - aggregating together many states with some common characteristics such as beyond a certain number of failures
  - then considering them to be first operational states to achieve lower bound, and then failed states to achieve the upper bound

Original Definition

- Assume the original MC is a stochastic process X={X(t); t>=0}, with state space $O \cup \{f\}$, where O is the set of operational states in which the system is up; f is a failed and absorbing state in which the system is down

$$UR_{system}(t) = P\{X(t) = f\}$$

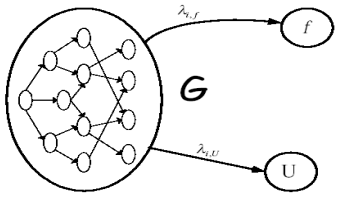
An **absorbing** state is a state that, once entered, cannot be left until the system starts a new mission

Bounding Methods

- We generate a new MC, X`={X`(t); t>=0}, with state space G $\cup$ {f, U}, where G is a set of operational states, which is a subset of O, f is a failed and absorbing state, U is a set of truncation states.
- In general, the states aggregated into U may include both operational states and failed states
- Then

$$[UR(t)]_{lb} = P[X'(t) = f],$$

$$[UR(t)]_{ub} = P[X'(t) \in \{u, f\}].$$



References

R. R. Muntz, E. de Souza e Silva, and A. Goyal, *Bounding availability of repairable computer systems*, IEEE Transactions on Computers **38** (1989), no. 12, 1714–1723.

E. de Souza e Silva and P.M. Ochoa, *State space exploration in Markov models*, Performance Evaluation Review **20** (1992), no. 1, 152–166.

J. C. S. Lui and R. R. Muntz, *Computing bounds on steady-state availablility of repairable computer systems*, Journal of the ACM **41** (1994), no. 4, 676–707.

P. Semal, *Refinable bounds for large Markov chains*, IEEE Transactions on Computers **44** (1995), no. 10, 1216–1222.

S. Mahévas and G. Rubino, *Bounding asymptotic dependability and performance measures*, 1996, pp. 176–186.

J. A. Carrasco, *Bounding steady-state availability models with group repair and phase type repair distributions*, Performance Evaluation **35** (1999), no. 4, 193–214.

Summary of Lecture #16

- Safety is the probability that a safe action will result after a failure occurs, highly dependent on the applications
- Safety analysis usually requires state space methods (e.g., Markov)
- Markov models are powerful in terms of modeling capabilities (repair, coverage, spare, dependencies, etc), but suffer from the state-explosion problems
- An efficient and accurate solution is to use modularization, which combines the best of Boolean (BDD) and Markov approaches
- Another solution is to use the bounding method to obtain an approximate estimate.

Things to Do

- ECE544 Project
  - Final report due **Wed., Nov. 30**
  - Presentation slides due **Mon., Dec. 5**
  - Please check out Report & Presentation Guidelines for requirements.

Next Topic

- Network reliability analysis