Chapter 5 Probabilistic Risk Assessment

5.1 Introduction

5.1.1 Chapter Content

This chapter presents a methodology for event analysis. DBAs, as discussed in the previous chapter, define the events to be analyzed for PSAs. Risk assessment is made possible by a piecewise refinement of the scenario, following each possible sub-sequence or branch. At each stage, fault tree analysis determines the branch probability. The overall probabilities of each possible end result can thus be calculated.

5.1.2 Learning Outcomes

The overall objectives for this chapter are as follows:

Objective 5.1	The student should be able to construct an event tree for simple examples and evaluate the event frequencies given the branch probabilities.					
Condition	Closed book written examination.					
Standard	100% on the methodology, evidence of good judgement on scenario development.					
Related concept(s)						
Classification	Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
Weight	a	a	a			a

Objective 5.2	The student should be able to construct a fault tree for simple examples and evaluate the top failure rate or probabilities.					
Condition	Closed book written examination.					
Standard	100% on the methodology, 75% on technical accuracy.					
Related concept(s)						
Classification	Knowledge	Comprehension	Application	Analysis	Synthesis	Evaluation
Weight	a	a	a			

5.1.3 The Chapter Layout

The exploration proceeds by first addressing event trees. The fault tree discussion then follows logically since event trees require the probabilities that come from fault trees.

5.2 Event Trees

5.2.1 Event Tree Methodology

Following [MCC81] event trees are constructed by starting with an initiating failure event and following it in time. As the event unfolds in time, various mitigating systems will be called into play. These mitigating systems (usually safety systems) will either deploy successfully for they will fail to deploy successfully. Partial deployment is possible but in Canada this possibility is discounted to be conservative. At each bifurcation, probabilities are assigned (usually with the help of fault trees) and each branch is followed systematically. Branch probabilities can be conditioned by the upstream states, that is, the proper functioning of a mitigating system might be a function of the event details such as whether another system failed or not earlier in the sequence. Whenever success or failure choices are not permitted for a system, the failure probability of that system being set equal to unity because of previous events. Figure 5.1 illustrates a typical event tree structure. Typically the ET is sequences to follow the event chronologically but that is not a requirement; there can be instances when the ET can be made simpler by reordering the branch sequence.

The ET provides the overall structure of the event, the Fault Trees (FT), discussed later in this chapter, provide the probabilities for the branches of the ET. The consequences of the various event paths are determined by analysis, as discussed in the next chapter. Analysis is often required to determine the event sequence as well since the scenario details often depend on the response of the process and safety systems during the event.

5.2.2 Event Tree Example: Collision Avoidance

A simple example is useful to illustrate the construction of an event tree (figure 5.2). Consider a motorist driving down a country road at night. A deer steps in front of the car. This is the initiating event (IE). The question is: "What are the possible outcomes and what is the probability for each outcome?". The usual procedure to answer this is to follow the event in time. First, does the driver see the deer? Yes or no? Let us assume that the probability of Yes is 0.9. Thus the probability of No is 0.1. We assume that there is no middle ground, the deer is seen or it is not seen. We could, if necessary, pose an intermediate case of the driver seeing something that is not yet identified (ie, there is definitely something there but it is not known whether it constitutes a hazard or not). Such an eventuality may prove important to the analysis or it may not. It is expedient to keep the first attempt at the ET construction simple and refine only if necessary.

If the driver sees the deer, does the driver apply the brakes in time? Yes or no? Let us assume the probability of a Yes is 0.8.

If the brakes are applied, do they work? Yes or no? Let us assume the probability of a Yes is 0.999 (mechanical systems are typically much more reliable than humans).

Finally, what does the deer do? Does the deer evade the car? Yes or no? Let us assume the probability of a Yes is 0.5.

We need to follow each possible combination of events. If the driver sees the deer, applies the brakes and the brakes work, then it doesn't matter whether the deer evades or not. So that path does not branch at the deer evasion question.

If the driver does not see the deer, then the brakes do not play a part in the scenario, but the deer evasion does.

If the driver sees the deer but does not apply the brakes in time, then whether the brakes work or not is immaterial. But the deer evasion is important.

If the driver sees the deer, applies the brakes but the brakes fail, then again, deer evasion is important.

The evaluation of the probability of each branch is relatively straightforward: Assuming the events are independent, we simply multiply the appropriate branch probabilities together, for instance, the chance of seeing the deer, applying the brakes, the brakes work and the deer is thus missed is 0.9 x 0.8 x .999 = 0.7193.

The most uncertain part in all this is assigning the correct probabilities. Depending on the purpose of the analysis, we might want to construct several ET's, one for each type of driver or each possible state of the driver (sleepy, experienced, etc.).

5.2.3 Event Tree Example: Large LOCA

[see example in appendix 6, an excerpt from a SAR, Wolsong-2 volume 1 chapter 5]

5.3 Fault Trees and Cut Sets

Simple systems are amenable to a state by state analysis as in the examples in chapter 2. But most systems are much more complex than that. Constructing a fault tree is a more systematic procedure that permits the analysis of complex systems. However, one major drawback is that redundant states arise that will lead to double accounting if they are not eliminated or "cut from the set", giving rise the term *Minimum Cut Set.* Minimum Cut Sets are formally defined as a set of events that, if they all occur, will cause system failure [MCC81].

To illustrate, consider a system fault tree with a top event composed of X AND Y. Event X and event Y are developed as shown in figure 5.3.

We note that event B occurs in two locations. To calculate the probability of the top event, we construct the Boolian expressions:

$$\begin{aligned} TOP &= X\,.\,Y \\ X &= A + X1 \\ Y &= D + Y1 \\ X1 &= B + C \\ Y &= B + E \end{aligned} \tag{1}$$

Therefore:

$$\begin{aligned} X &= A + B + C \\ Y &= D + B + E \end{aligned} \tag{2}$$

and, thus:

$$\begin{aligned} TOP &= (A+B+C).(D+B+E) \\ &= AD+AB+AE+BD+BB+BE+CD+CB+CE \end{aligned} \tag{3}$$

This last expression can be simplified because:

$$B + B\,.\,\text{anything} = B \tag{4}$$

Thus the final expression is the minimal cutset:

$$\begin{aligned} TOP &= B+AE+CD+CE \\ &= B+(A+C).(D+E) \end{aligned} \tag{5}$$

The revised fault tree is shown in figure 5.4. To illustrate the importance of generating the minimum cutset, consider the case where the probability of events A, B, C, D and E are each 0.01. Equation 5 (the minimum cutset) gives a TOP event probability of 0.0104 whereas equation 3 yields 0.0009; this is quite a difference.

For anything but the simplest fault trees, it is far too cumbersome to follow the above procedure to generate the minimum cut set. However, it is useful to survey actual fault trees looking for redundancies and interdependencies since this exercise offers some insight into the system behaviour. The general rule is to look for repeated events (B in this case). Trace to the top event looking for ANDing of the repeated event. If ANDing occurs, the tree cannot be simplified with respect to the repeated event. If ANDing does not occur, then the tree is simplified by making B an independent event. Practically, however, a computer program is required to analyze actual fault trees.

5.4 Fault Tree Data

5.4.1 Equipment Reliability Data

[see excerpts from MCC81 in appendix 7]

5.4.2 Human Reliability Data

[see excerpts from MCC81 in appendix 7]

5.5 Exercises

1. Construct an event tree for the deer evasion example given in section 5.2 but this time put the “deer evades” branch first. Is the resulting ET simpler? Have the total Hit and total Miss probabilities changed?

2. Generate a fault tree for the shutdown safety system of a small research reactor such as MNR:

3. For a small research reactor such as MNR
   a. Generate a Large LOCA event tree.
   b. Generate the branch probabilities using fault trees if necessary.

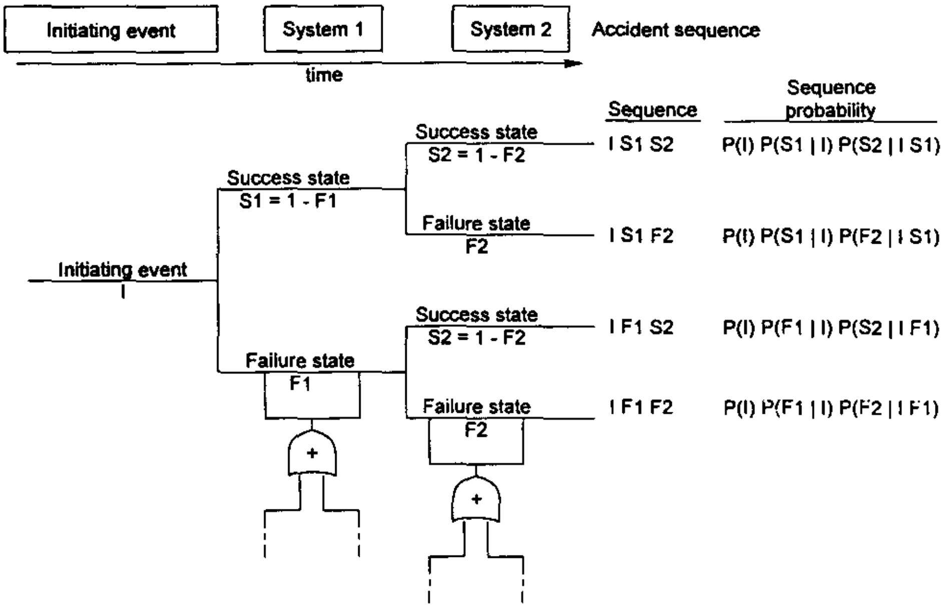

Figure 5.1 Event Tree Illustration [MCC81]

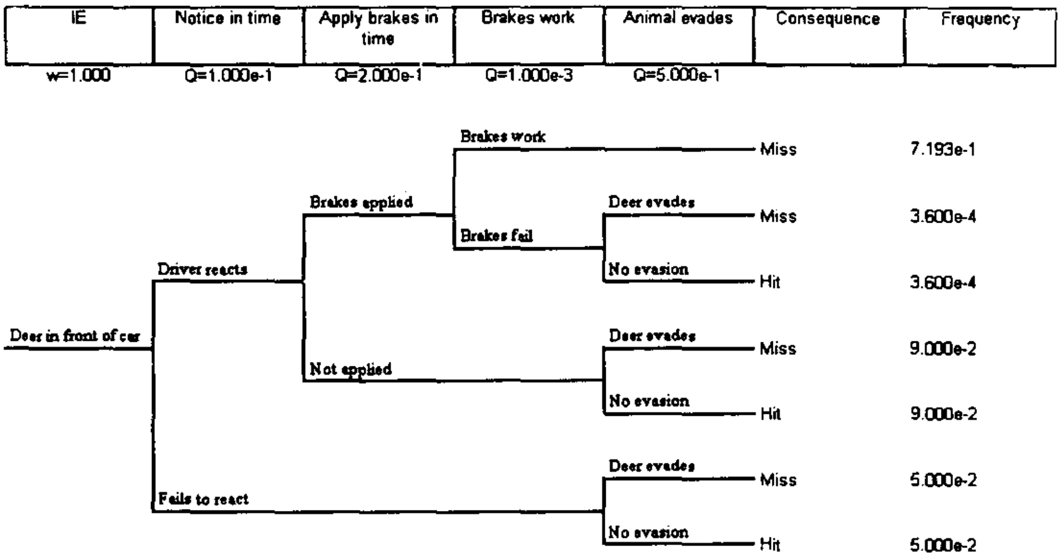

Figure 5.2 Example event tree - Avoiding a deer

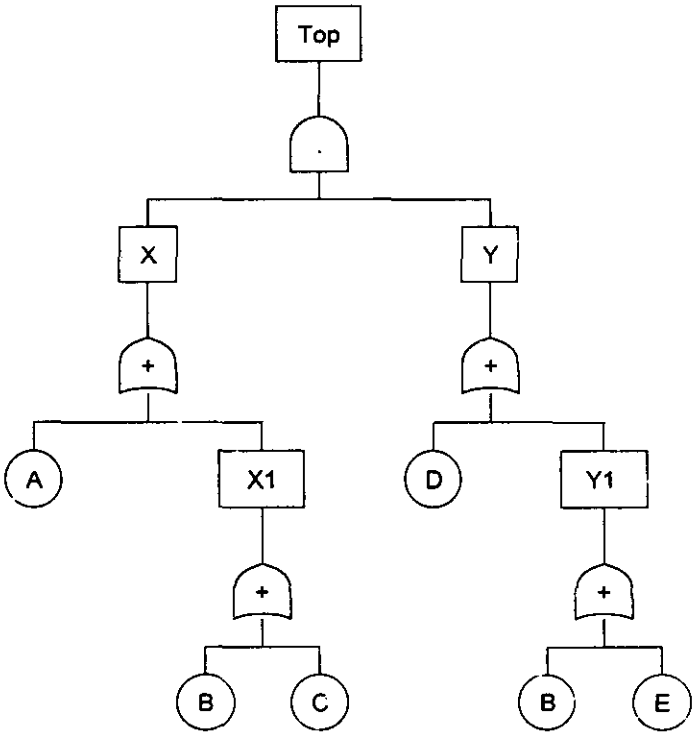

Figure 5.3 Fault tree example

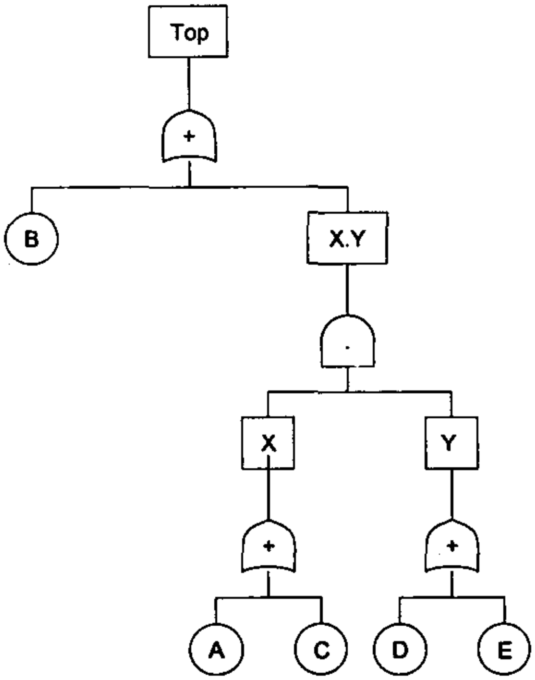

Figure 5.4 Fault tree - minimum cut set