ÁRVORES DE FALHA

Uma árvore de falha é um diagrama de seqüência de eventos que permite através de lógica dedutiva aplicada de trás para diante chegar-se às causas-raiz de uma dada falha. Os eventos são organizados em uma estrutura lógica que utiliza "portas lógicas" para identificar a relação causal entre os eventos imediatamente abaixo da "porta lógica". Árvores de falha podem ser utilizadas na determinação dos ramos que irão compor uma árvore de eventos.

Os procedimentos para construir árvores de falha são bem documentados e constituem-se em um instrumento para o entendimento de inter-relações complexas.

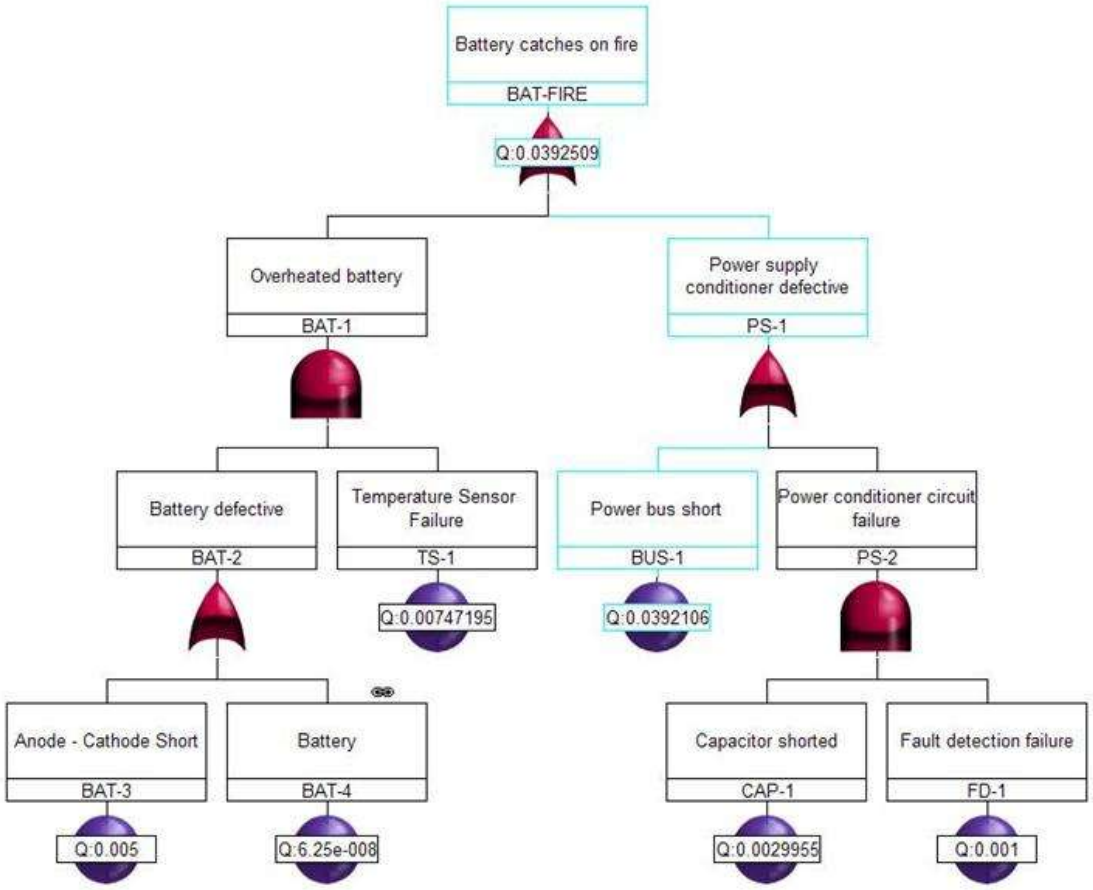
A construção de uma árvore de falha requer um conhecimento íntimo do sistema que está sendo estudado, identificando as causas, determinando a unicidade, a independência e a condicionalidade dos eventos envolvidos. A descrição lógica da interações entre eventos requer considerável esforço de reflexão e entendimento. Este processo é, entretanto, uma das partes as mais valiosas do processo da análise.

As árvores de falha são análogas a uma fotografia que descreve circunstâncias em um instante a tempo ou uma transição entre dois eventos consecutivos em uma árvore de eventos. As árvores de falha podem ser usadas qualitativamente para identificar os eventos que causam a falha. Quando as probabilidades são determinadas para os eventos, a probabilidade da falha do evento principal pode ser calculada. Os relacionamentos dependentes do evento do tempo não podem ser representados em uma árvore de falha.

Variações da análise da árvore de falha podem ser exploradas, focalizando em ações que impedirão o progresso para uma falha total. "Árvores do sucesso" e diagramas mais gerais têm em comum com árvores de falha a característica de ilustrar graficamente a cadeia de eventos, incluindo causas e conseqüências, que podem levar de um dado evento inicial tanto à perda total de um equipamento quanto ao seu desempenho seguro e aceitável.

Relex Corporation

http://www.relex.com/



Basic Event

A **basic event** represents the lowest level, or terminating event, in a fault tree. A basic event has no inputs. It simply represents the occurrence of an event in the system being analyzed. A basic event is either a component-level event that is not further resolved or an external event. Basic events can include hardware or software failures, human errors, and system failures. Basic events are the most commonly used primary events in FTA. Primary events are lower-level events.

Basic Event

Undeveloped Event

An **undeveloped event** is used if further resolution of that event is not necessary for proper evaluation of the fault tree or if the information necessary for developing this event is not currently available. While an undeveloped event is similar to a basic event, it is represented by a different symbol to signify that it is possible to break this event down into associated gates and events, even though this has not been done for this particular analysis.

Undeveloped Event

Conditional Event

A **conditional event** is used along with an Inhibit gate, which is described later in this article. In a fault tree with an Inhibit gate, the output occurs only when the input events occur AND a conditional event is satisfied.

Conditional Event

House Event

A **house event** is a special type of event employed for specific uses within a fault tree analysis. Common uses for house events are:

- To represent an event that is normally expected to occur.
- To disable or enable parts of a fault tree to make them functional or non-functional.
- To represent trigger events, switching events, and external events.

House Event

AND Gate

The **AND gate** is used to indicate that the output occurs if and only if all the input events occur. There must be at least two input events to an AND gate.

AND Gate

OR Gate

The **OR gate** is used to indicate that the output occurs if and only if at least one of the input events occur. There must be at least two inputs to an OR gate.

OR Gate

Voting Gate (m/n)

The **Voting (m/n) gate** is used to indicate that the output occurs if and only if at least *m* out of the *n* input events occurs. The input events do not need to occur at the same point in time but should be present at the same time. The output occurs when at least *m* input events occur. When *m* = 1, the Voting gate behaves like an OR Gate.

Voting Gate

Summary of Logic: If *m* = 2 and *n* = 3, two input events must be TRUE for the output to be TRUE. If zero or one input events are TRUE, the output is FALSE.

A truth table for a Voting (2/3) gate follows. The Boolean equation for a three-input Voting gate is T = (A * B) + (B * C) + (C * A).

A	B	C	Output
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	F

Inhibit Gate

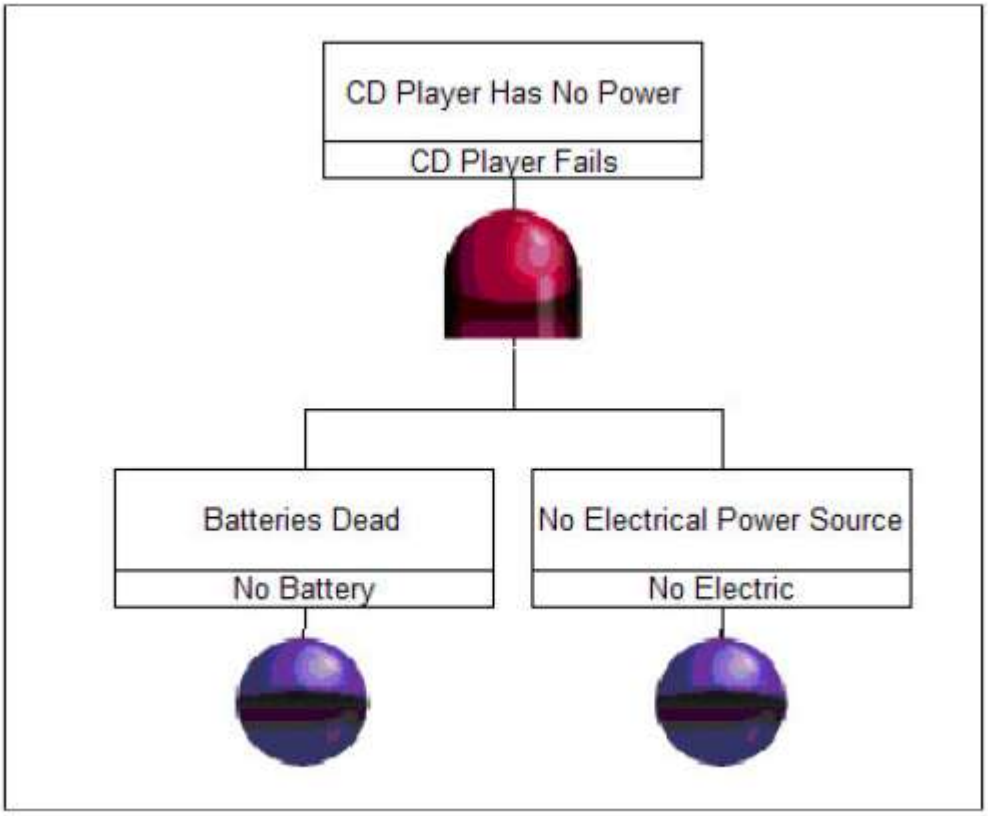
The **Inhibit gate** is used to indicate that the output occurs when the input events (I1 and I2) occur and the input condition (C) is satisfied. An Inhibit gate is very much like an AND gate with a condition. When you insert an Inhibit gate in a Relex fault tree, a conditional event is automatically inserted along with it so that the input condition can be specified.

Inhibit Gate

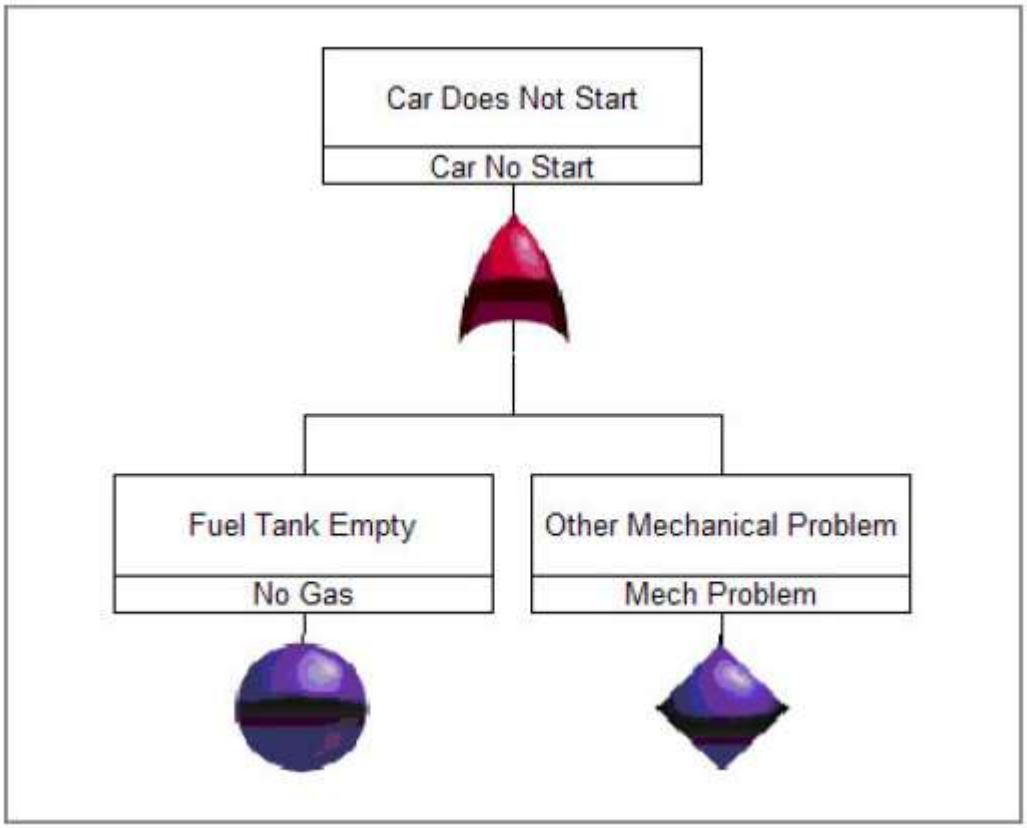
Summary of Logic: If one input event is FALSE, the output is FALSE. If all input events and the input condition are TRUE, the output is TRUE.

A truth table for an Inhibit gate follows. The Boolean equation for an Inhibit gate is T = (I1,I2) * C.

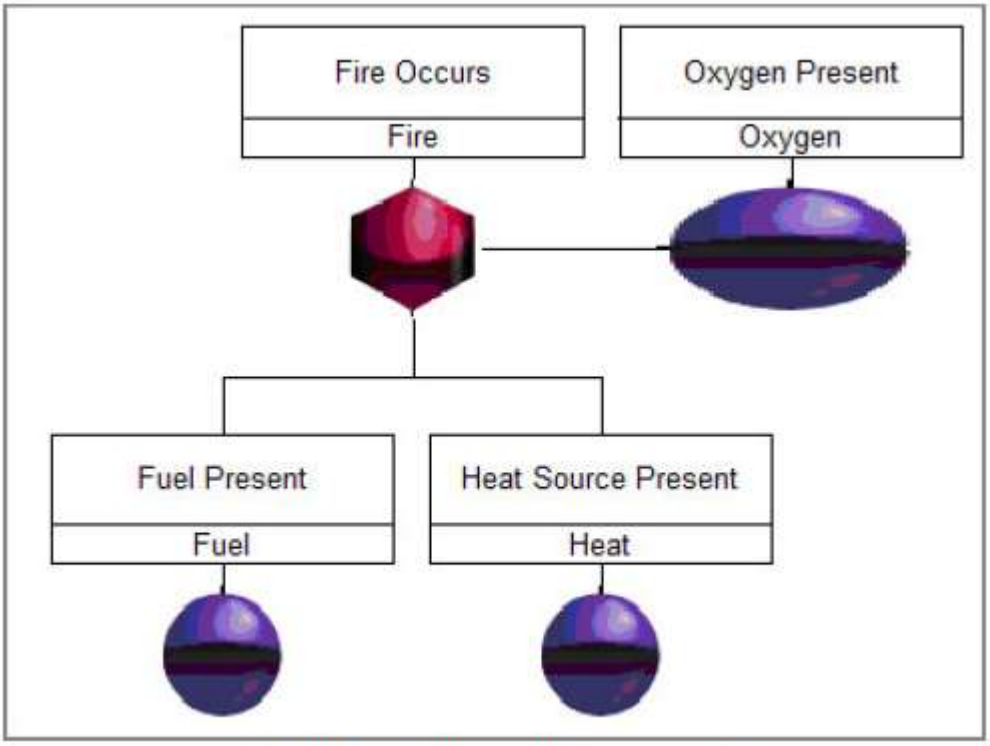
I1	I2	C	Output
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	F
F	T	T	F
F	T	F	F
F	F	T	F
F	F	F	F



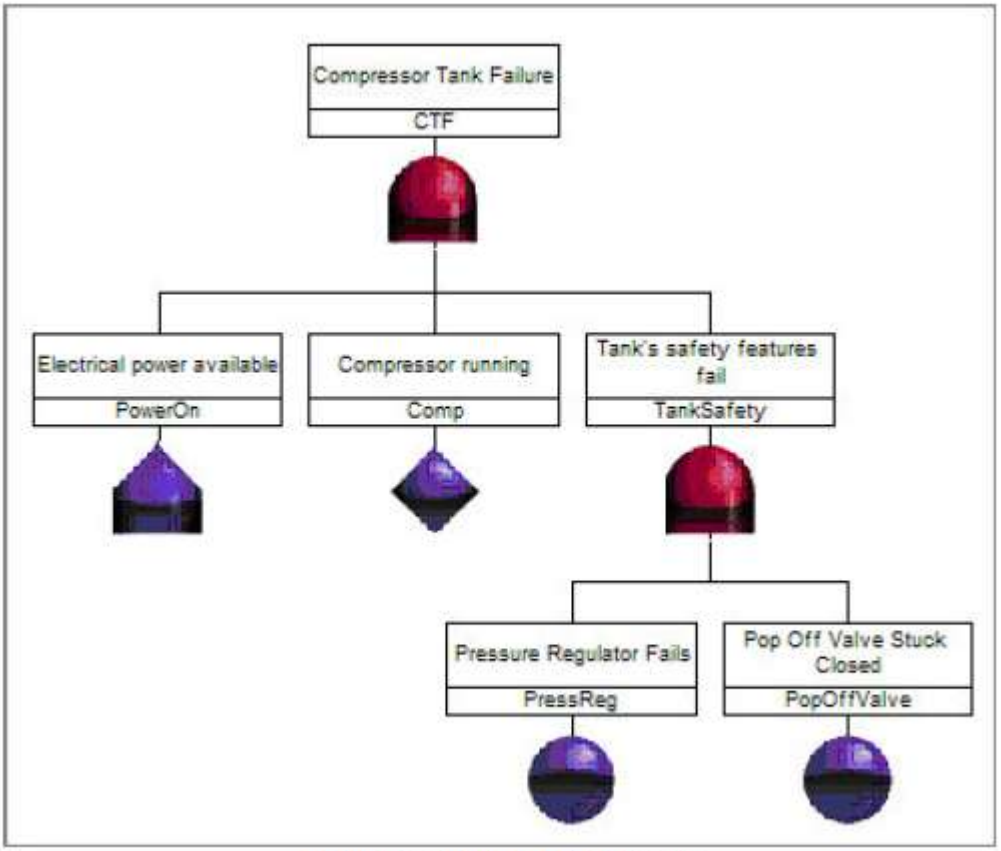
Fault Tree with Two Basic Events



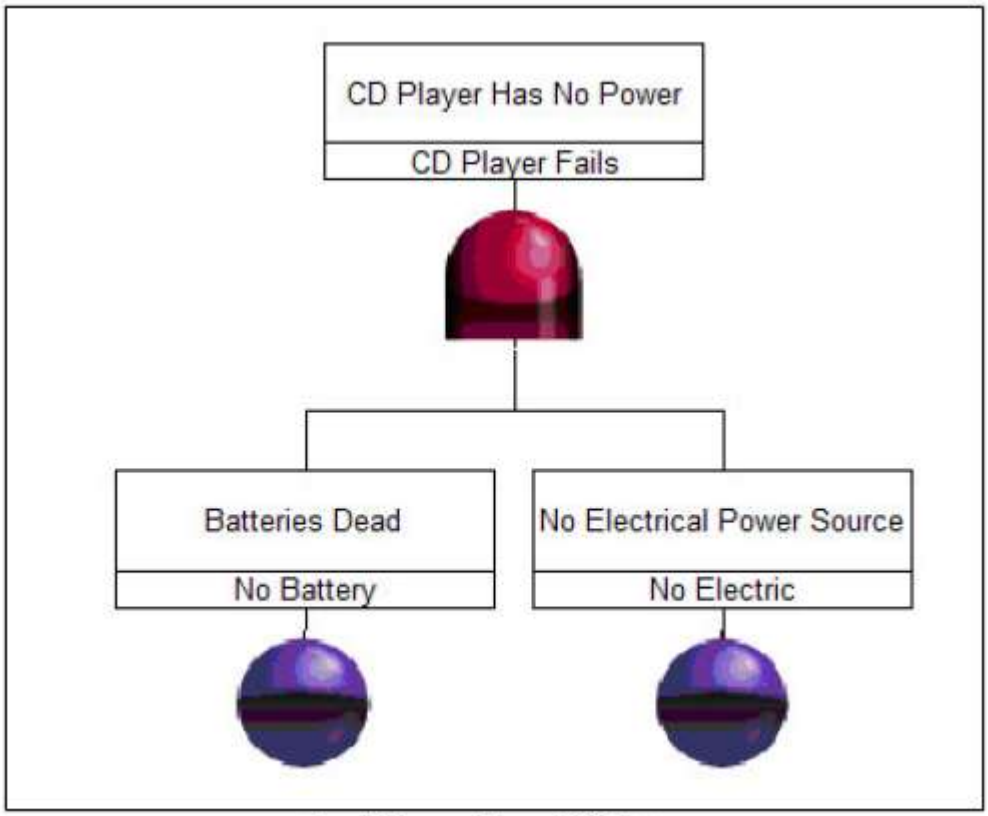
Fault Tree with an Undeveloped Event



Fault Tree with a Conditional Event

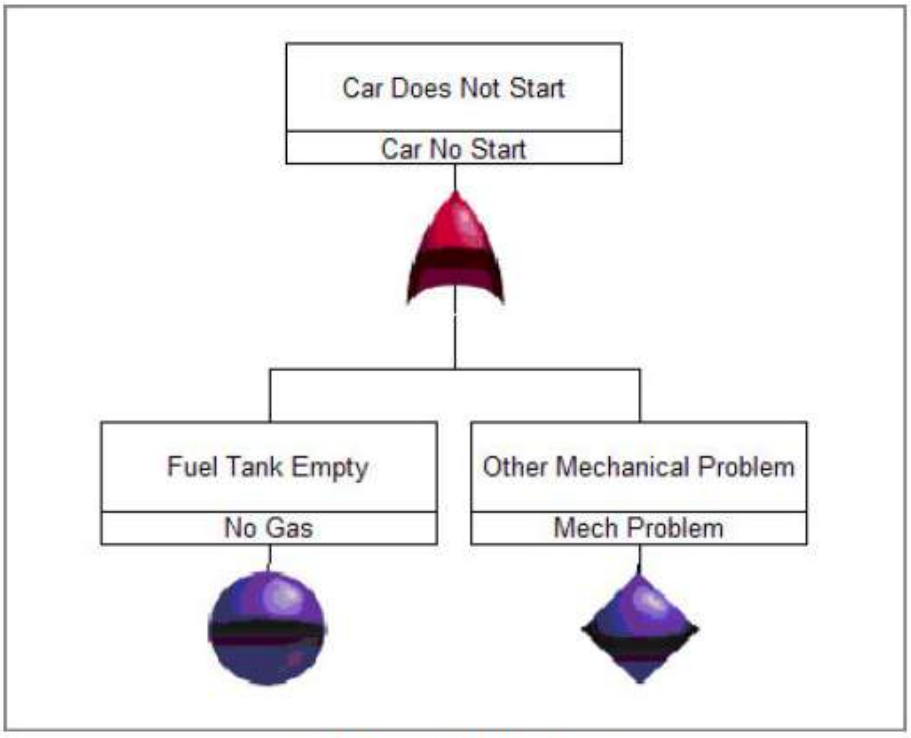


Fault Tree with a House Event



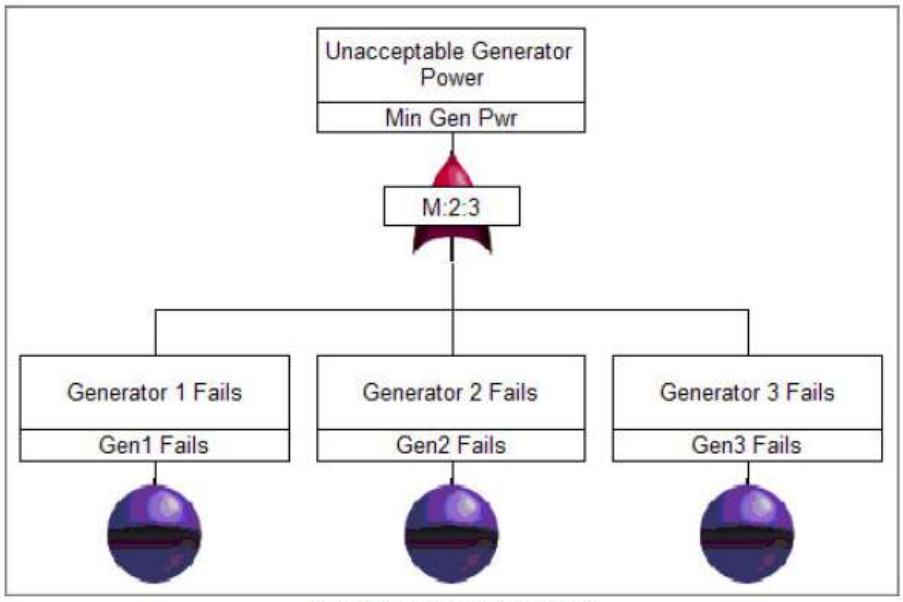
Fault Tree with an AND Gate

A vehicle will not start if there is no fuel OR there is some other mechanical problem.



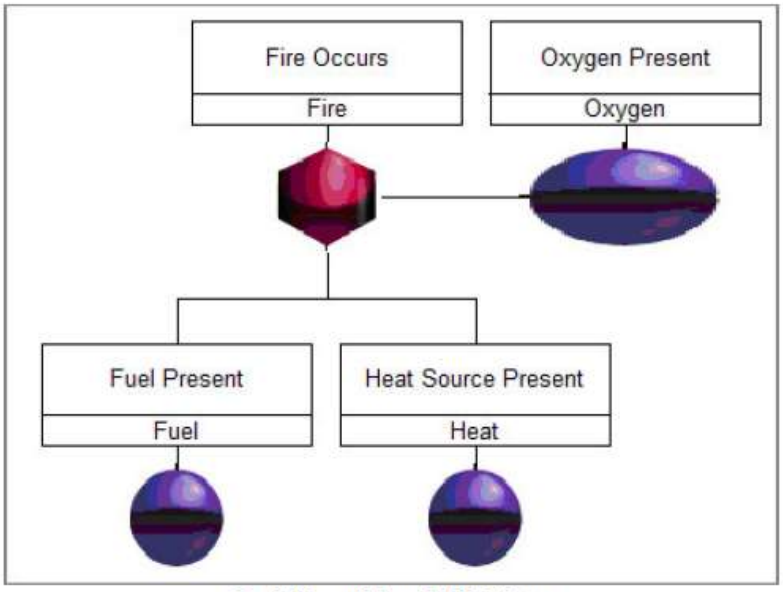
Fault Tree with an OR Gate

Power is supplied by three generators. When two generators are working, adequate power is supplied. When only one generator is working, there is not enough power.



Fault Tree with a Voting Gate

From the fire triangle, we know that there will be fire when fuel and a heat source are present, given the presence of oxygen. The two basic events and the conditional event are the three legs of the triangle. All three events must be TRUE for the top event (Fire) to occur.



Fault Tree with an Inhibit Gate