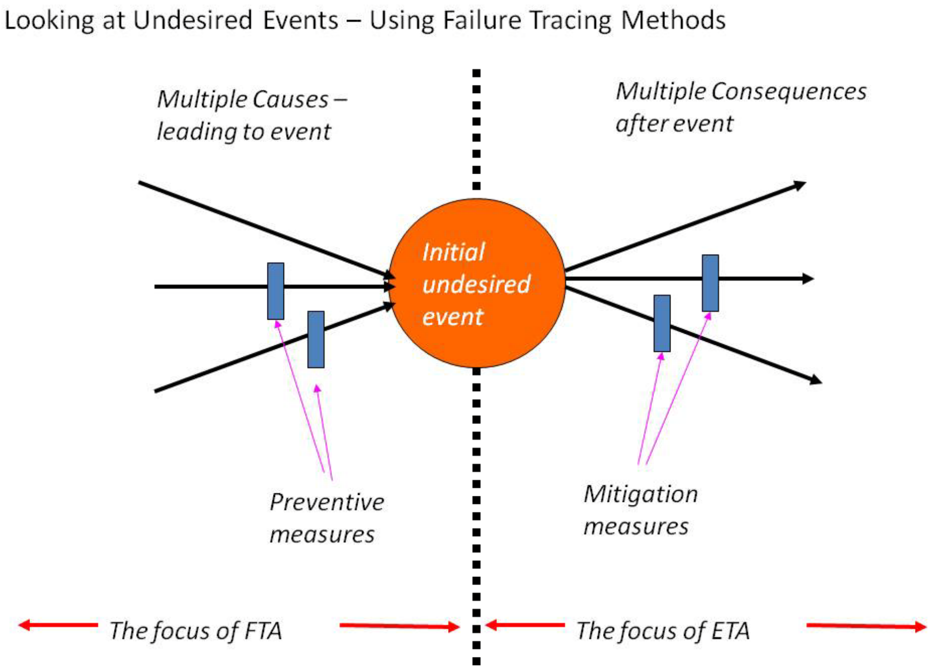
Fault Tree Analysis (FTA) and Event Tree Analysis (ETA)

It is easy to get confused between these two techniques. Indeed, the two are in fact complimentary (and are often used together) but focus on opposite sides of an undesired event. The diagram below shows how they fit together:

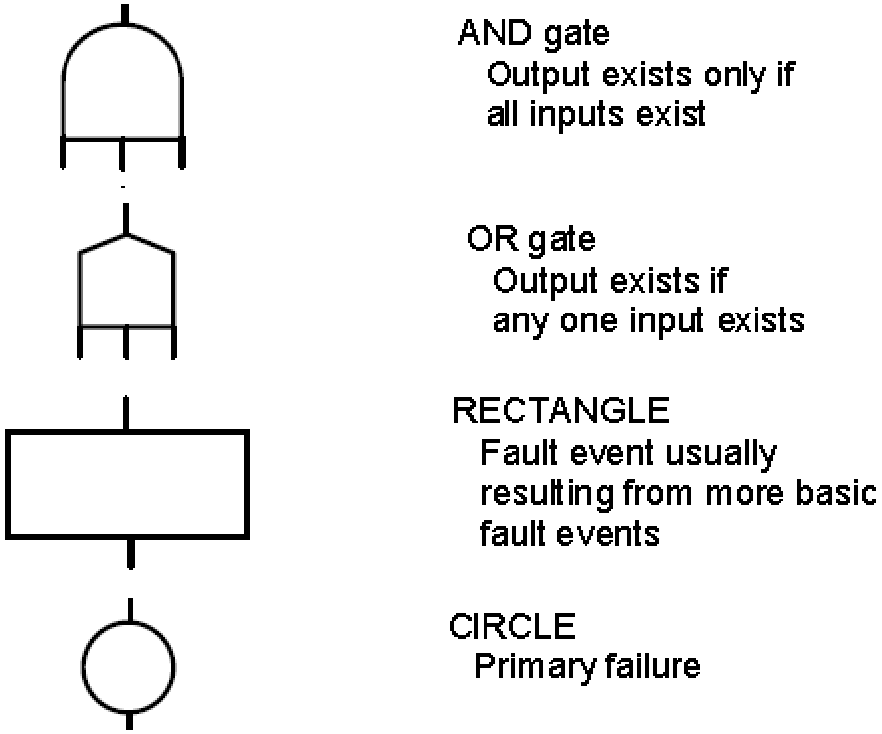

Looking at Undesired Events – Using Failure Tracing Methods
Multiple Causes – leading to event
Multiple Consequences after event
Initial undesired event
Preventive measures
Mitigation measures
The focus of FTA
The focus of ETA

This is sometimes called a 'bow-tie' model (because it looks like one) and when complimentary FTAs and ETAs are used, it's called the bow-tie technique. The diagram only shows a single 'undesired event'; in reality, multiple causes can lead to many different events initially, each then escalating with multiple consequences. You can analyse each event with FTA and ETA. In summary, FTA is concerned with analysing faults which might lead to an event, whereas ETA is interested in stopping it escalating. Both can be applied qualitatively or, if you have the data, quantitatively.

Fault Tree Analysis (FTA)

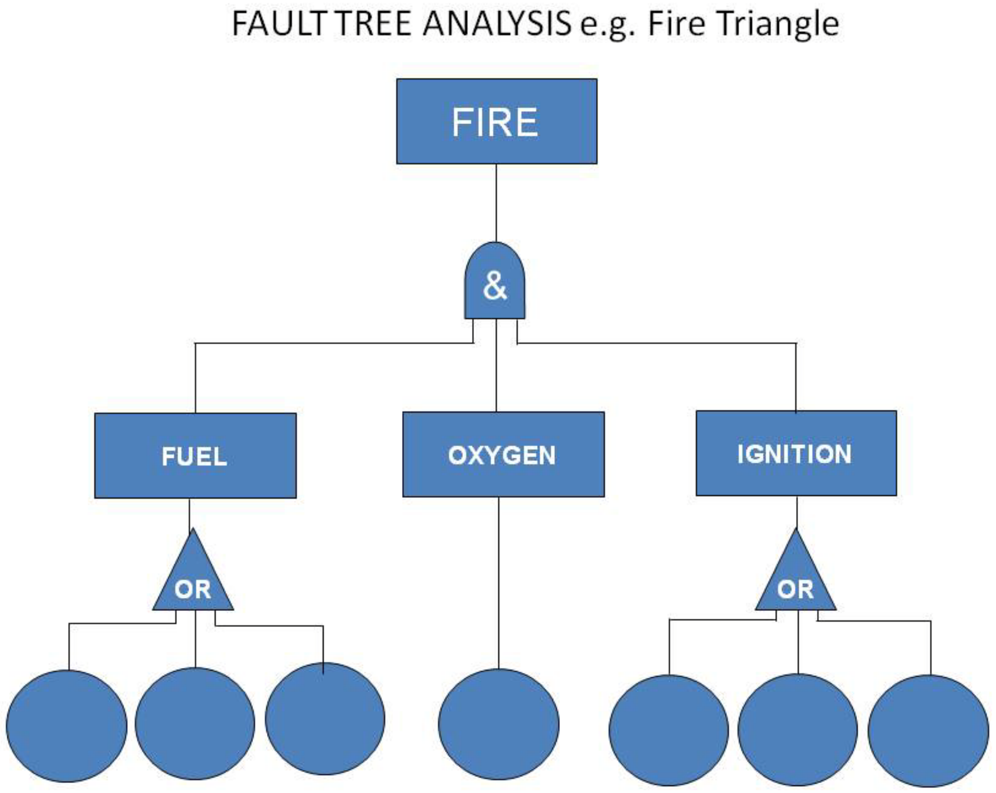
In many cases there are multiple causes for an accident or other loss-making event. Fault tree analysis is one analytical technique for tracing the events which could contribute. It can be used in accident investigation and in a detailed hazard assessment.

The fault tree is a logic diagram based on the principle of multi-causality, which traces all branches of events which could contribute to an accident or failure. It uses sets of symbols, labels and identifiers. But for our purposes, you'll really only use a handful of these, shown below:


AND gate
Output exists only if all inputs exist
OR gate
Output exists if any one input exists
RECTANGLE
Fault event usually resulting from more basic fault events
CIRCLE
Primary failure

Even then, you'll see variations of these symbols in use.

A fault tree diagram is drawn from the top down. The starting point is the undesired event of interest (called the 'top event' because it gets placed at the top of the diagram). You then have to logically work out (and draw) the immediate contributory fault conditions leading to that event. These may each in turn be caused by other faults and so on. It could be endless (though, in fact, you will naturally have to stop when you get as far as primary failures). The trickiest part of the whole thing is actually getting the sequence of failure dependencies worked out in the first place. Let's look at a simple example to illustrate the point.



The above figure shows a simple fault tree for a fire.

EXAM HINT: Don't worry about getting the symbols precisely right when drawing fault trees by hand; you can make your intentions quite clear enough simply by writing AND or OR in the appropriate logic gate as well. Similarly, so long as you describe the fault/failure in a box, don't worry unduly about the (sometimes subtle) distinction between what should go in rectangles and circles.

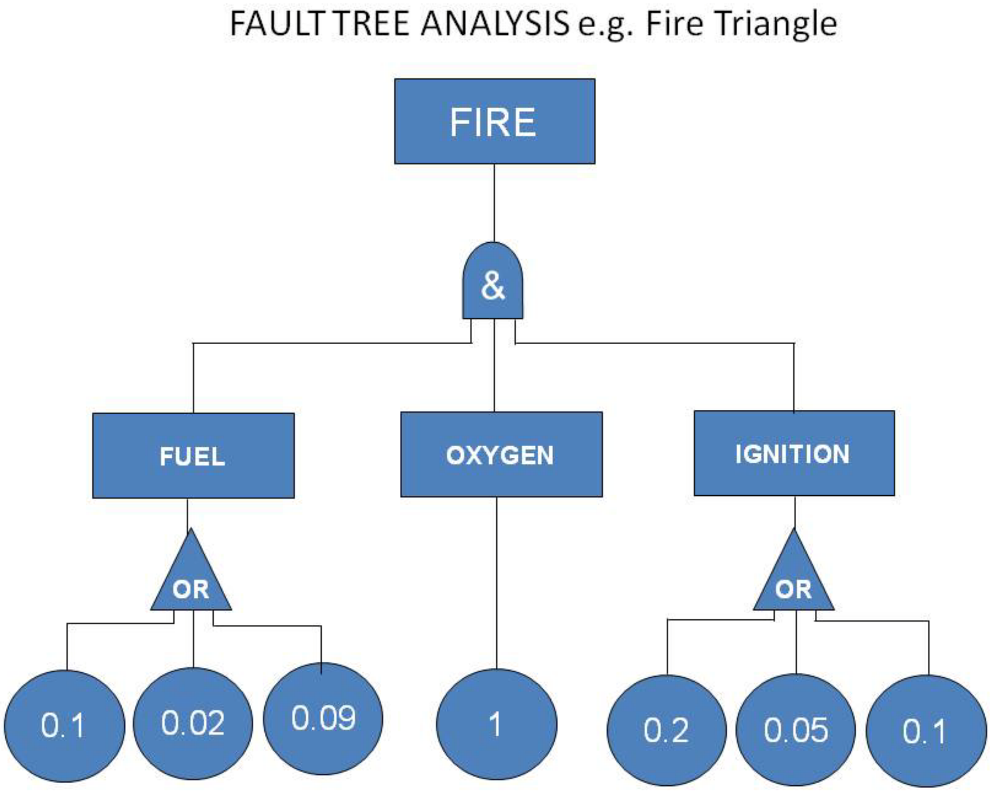
For the fire to occur there needs to be:

- Fuel.
- Oxygen.
- An ignition source.

Notice we use an AND gate to connect them here because all three need to be present *at the same time* to allow the top event. The example shows that, in this scenario, there happen to be three possible sources of fuel and three possible sources of ignition. An OR situation applies in each case, because it would only need one of these to be present. The example also shows a single source of oxygen (e.g. the atmosphere).

In order to prevent the loss taking place, we would first examine the diagram for AND gates. This is because the loss can be prevented if just one of the conditions is prevented.

Fault trees can also be quantified. Let's try this on the same example. From previous experience, or as an estimation, a probability for each of the primary failures being present or occurring can be established, shown below (these are purely illustrative):



(We've assigned a probability of 1 for Oxygen being present, as it always is in the surrounding atmosphere).

We can then use two well-established rules of combination of these probabilities and progress up the diagram to get at the probability of the top event (fire) occurring. Essentially we:

- *Add* the probabilities which sit below an OR gate (this isn't strictly correct, but is a 'rare event' approximation).
- *Multiply* the probabilities which sit below an AND gate

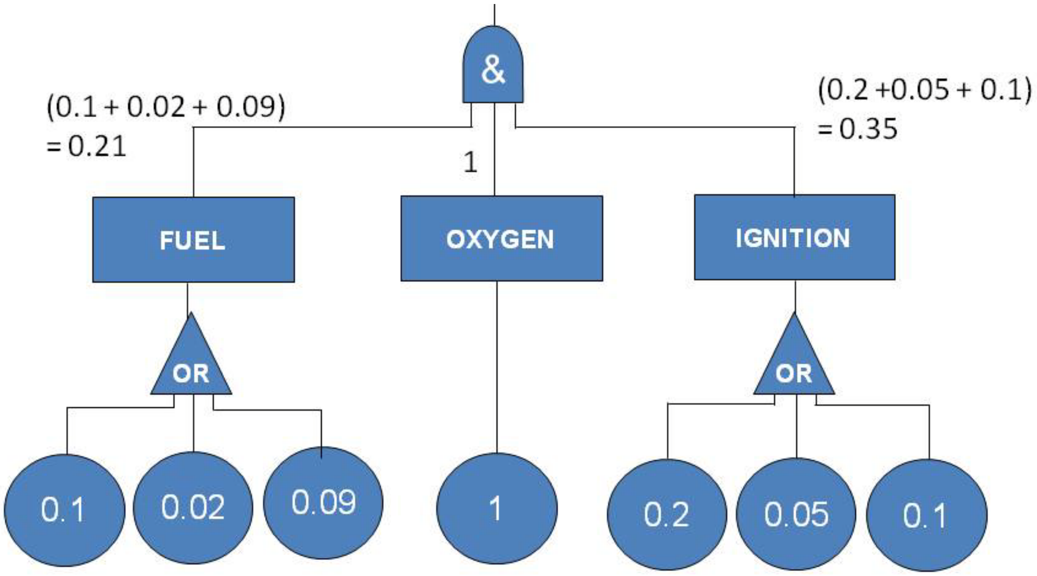
So, in this example, combining probabilities upwards to the next level gives:

Probability of FUEL being present = 0.1 + 0.02 + 0.09 = 0.21

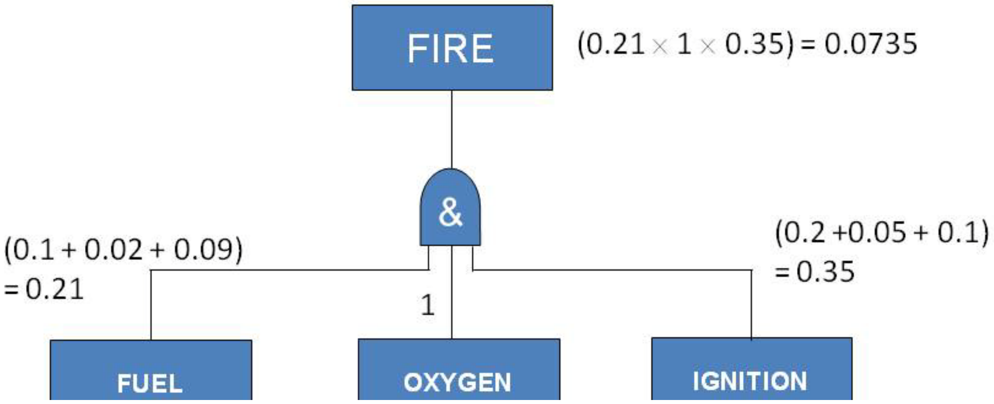
Probability of OXYGEN being present = 1

Probability of IGNITION being present = 0.2 + 0.05 + 0.1 = 0.35

Updating the diagram (and only showing the relevant part):

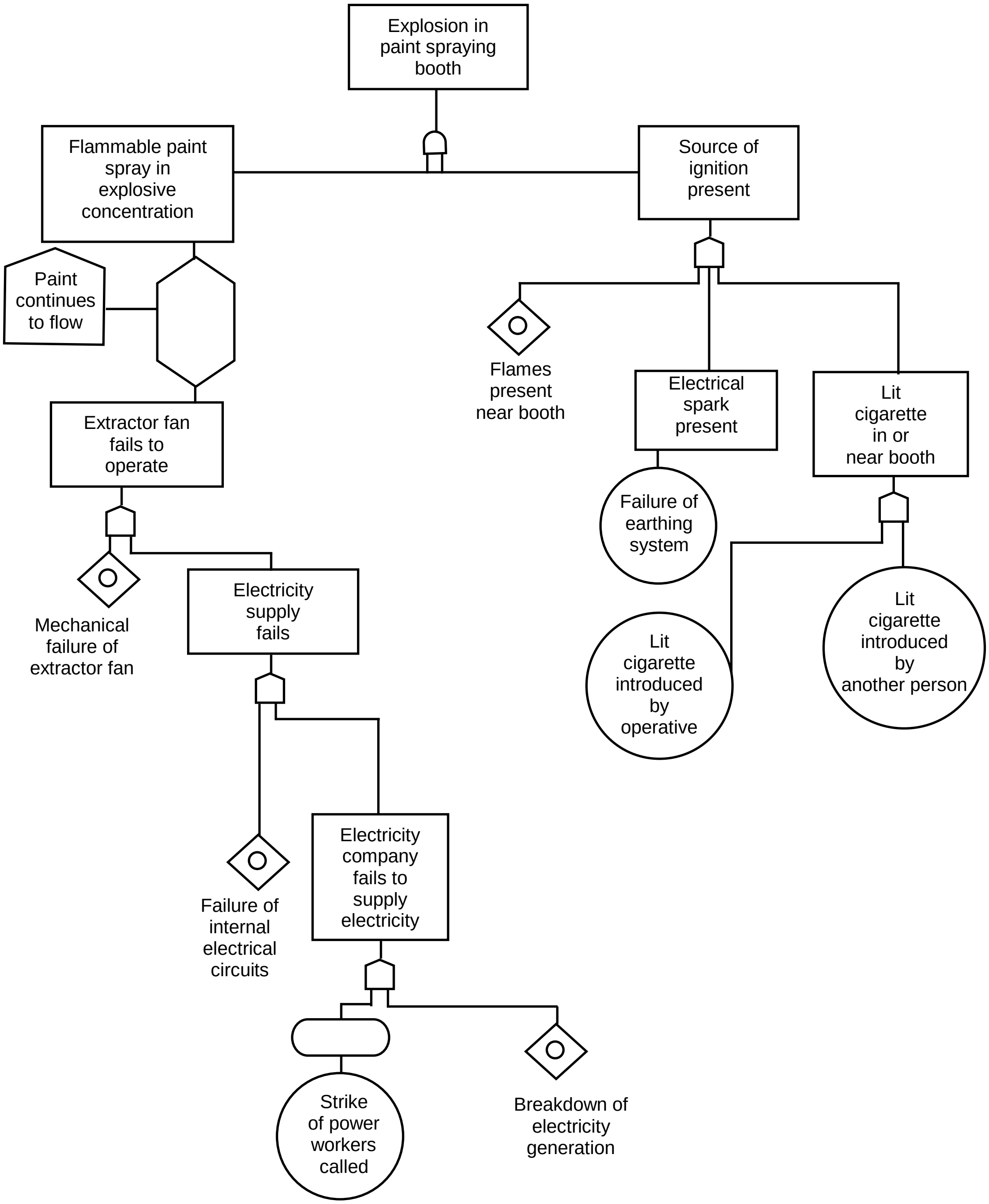


Moving up again, we can now calculate the probability of the top event. These faults are below an AND gate, so we multiply the probabilities, giving 0.21 x 1 x 0.35 = 0.0735. The top of the fully quantified fault tree then looks like this:



Exam hint: to gain maximum marks, make sure that you show all your working when quantifying a fault tree.

The above was a simple example. Below is another, more complicated fault tree (not quantified). It uses a much wider range of symbols too. Note that each route is developed until it ends at a primary event, or to a point where we consider that we have reached an equivalent situation. You should be able to make some possible improvements to this fault tree. The author seems to think that smoking is a major cause of ignition. There are probably many more examples of ignition causes that have not been considered. If you are interested in further reading on this subject, you might like to refer to BS EN 61025:2007 *Fault Tree Analysis*.


Explosion in paint spraying booth
Flammable paint spray in explosive concentration
Source of ignition present
Paint continues to flow
Flames present near booth
Electrical spark present
Lit cigarette in or near booth
Extractor fan fails to operate
Failure of earthing system
Mechanical failure of extractor fan
Electricity supply fails
Lit cigarette introduced by operative
Lit cigarette introduced by another person
Failure of internal electrical circuits
Electricity company fails to supply electricity
Strike of power workers called
Breakdown of electricity generation

Simplified Fault Tree for the Top Event: "Explosion in Paint Spraying Booth"

Event Tree Analysis (ETA)

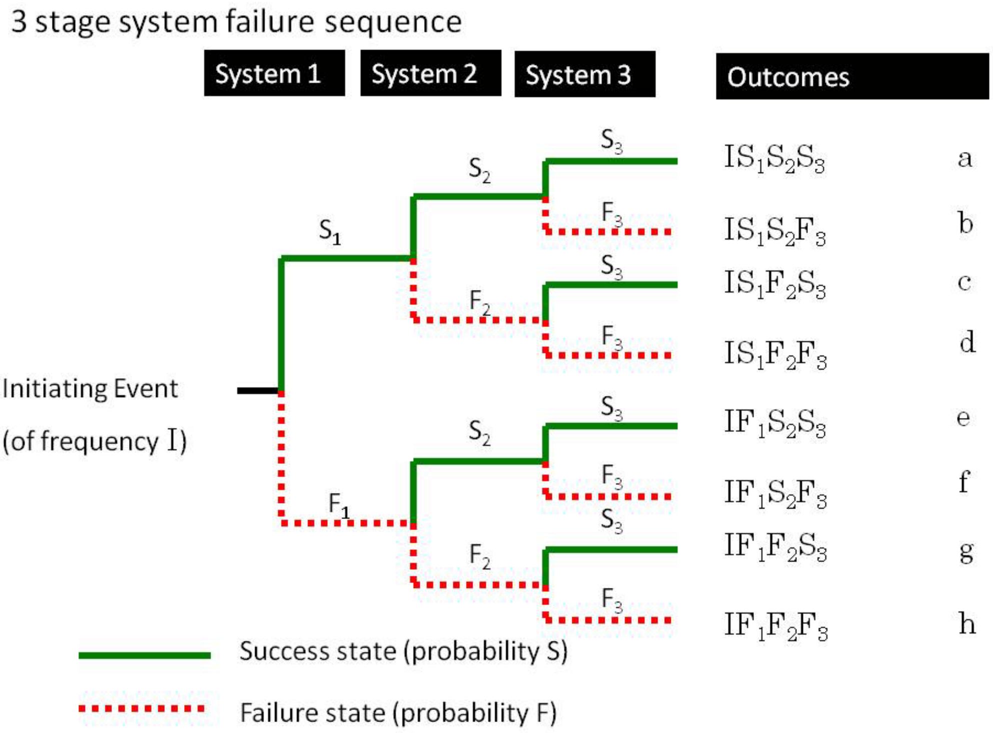
This is a complimentary technique to FTA but defines the consequential events which flow from the primary ‘initiating’ event. Event trees are used to investigate the consequences of loss-making events in order to find ways of mitigating, rather than preventing, losses.

Stages in carrying out event tree analysis:

1. Identify the primary event of concern.
2. Identify the controls that are assigned to deal with the primary event such as automatic safety systems, alarms on operator actions.
3. Construct the event tree beginning with the initiating event and proceeding through failures of the safety functions.
4. Establish the resulting accident sequences.
5. Identify the critical failures that need to be addressed.

There are a number of ways to construct an event tree. They typically use Boolean (or binary) logic gates, i.e. a gate that has only two options such as success/failure, yes/no, on/off. They tend to start on the left with the initiating event and progress to the right, branching progressively. Each branching point is called a node. Simple event trees tend to be presented at a system level, glossing over the detail.

The following is a generic example of how they can be drawn:

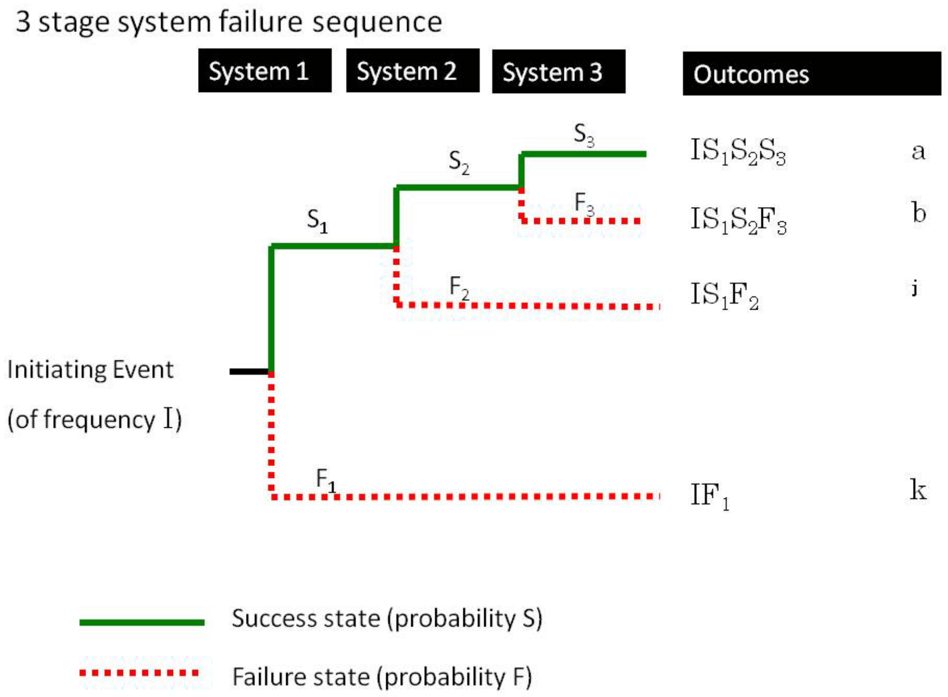


The diagram shows an initiating event (e.g. fire) and the subsequent operation or failure of three systems (e.g. fire suppression) which would normally operate should the event occur. Each system can either operate or not (somewhat unrealistic, as in some cases, things may partially operate). Because of the multitude of combinations of success/failure of each system, there are multiple possible final outcomes (labelled *a* to *h* in the diagram). The diagram also illustrates the way event trees can be quantified. The initiating event is typically specified as an expected annual frequency (e.g. 2 times per year) and the success/failure for each system as a probability.

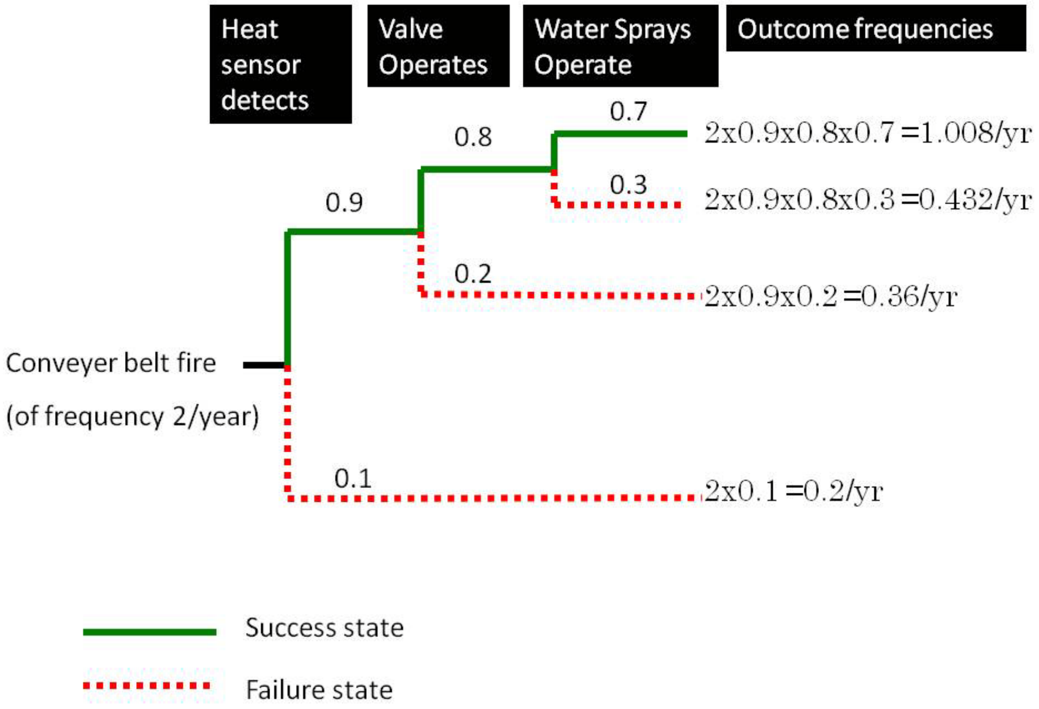
There are some rules to bear in mind when doing calculations:

- To calculate the frequency of each final outcome (labelled *a* to *h* in the diagram), you multiply along the branches, travelling from left to right from initiating event to final outcome. Thus, from the diagram, the frequency of the initiating event happening AND systems 1 AND 2 AND 3 working properly is I x S_1 x S_2 x S_3.
- The sum of each success/failure probability pair, at each specific node adds up to 1. So, for example, $S_1 + F_1 = 1$. This means that if you are only given the value for success probability of a particular system, it is easy to calculate the failure probability for that same system, because the two will add up to give 1.
- The sum of all the final outcome frequencies (for each outcome *a* to *h*) will add up to equal the frequency of the initiating event (it's bound to if you think about it).

This may sound rather complicated but, thankfully, event trees can often be simplified. For example, in situations where the failure of system 1 means that systems 2 and 3 will not stop a 'failure' outcome, there's no point in further branching along the failure branch of system 1. So, in the diagram, if the only real outcome that matters for things not to go out of control is for systems 1, 2 and 3 all to work properly (success), then everything else leads to uncontrolled consequences (to different degrees). Taking this approach, the above diagram can be drastically 'pruned', but the same calculation rules already outlined above apply:



To make this a little more real, the following diagram shows a quantified event tree for the action following a fire on a conveyor system.



The only outcome resulting in control of the event is where the sensor, valve and water spray operate (the example is a little contrived but serves to demonstrate the principles).

Notice how the frequencies of the outcomes are calculated. Notice also that the sum of all the outcome frequencies adds up to 2 in this case, i.e. the frequency of the initiating event (the conveyer belt fire).

The event tree could be used to check that there were adequate fire detection, warning and extinguishing systems.