How to Avoid the Generation of Circular Logic in a FTA

Ho-Gon Lim[a*], Sang-Hoon Han [a], Joon-Eon Yang [a]
[a]*Korea Atomic Energy Research Institute., Division of Integrated Safety Assessment, 1045 Daedeuk -Daero, Yusung-Gu, Daejeon, The Republic of Korea*
[*]*Corresponding author: hglim@kaeri.re.kr*

1. Introduction

An occurrence of a Circular Logic (CL) has been a problematic issue in a Fault Tree Analysis (FTA) [1-3].

In the present paper, we suggest the treatment criteria and method of the CL.

The main idea is to use the initial condition of a system before its failure. Depending on the initial condition of a system, it is shown that the CL should be treated differently.

2. CL and its initial condition

When a system is modeled by a mathematical way to predict the evolution of the status of the system, it is necessary to know the initial state of the system. However, in a fault tree analysis of a system, the initial state of the system is usually not used and an indefinite initial state is assumed to be assigned to the system for the whole consideration of the system's feasible failure situations. The ignorance of initial conditions in the FTA does not usually invoke a critical problem if a system' failure mode is definitely identifiable in connection with any other supporting system. However, if a CL is generated by coupling with other supporting systems, the initial condition of the system may be important factor to identify the treatment criteria of a CL. It is shown that, when the initial condition is considered for the individual failure sequence, the CL can be properly treated by reflecting the physical condition of a system.

In the following Sections 2.1, the treatment method of a CL generated by two systems interrelated each other are discussed with the generalization of the treatment methodology in multiple systems in Section 2.2.

2.1 CL in two interrelated systems

For the simple illustration of the treatment of CL depending on its initial condition, let's suppose a CL with two interrelated system. The simple Boolean relations of two systems with mutual dependencies have the following form

$$S_1 = c_1 + a_{12} \cdot S_2 \qquad (1)$$

$$S_2 = c_2 + a_{21} \cdot S_1 \qquad (2)$$

To eliminate the term, S2 in Eq. (1), Eq. (2) is substituted into Eq. (1) as follows:

$$\begin{aligned} S_1 &= c_1 + a_{12} \cdot (c_2 + a_{21} \cdot S_1) \\ &= c_1 + a_{12} \cdot c_2 + a_{12} \cdot a_{21} \cdot S_1 \end{aligned} \qquad (3)$$

To remove the term, S1, in Eq. (3), Eq. (2) is substituted into Eq. (3) as follows:

$$\begin{aligned} S_1 &= c_1 + a_{12} \cdot c_2 + a_{12} \cdot a_{21} \cdot (c_1 + a_{12} \cdot S_2) \\ &= c_1 + a_{12} \cdot c_2 + a_{12} \cdot a_{21} \cdot S_2 \end{aligned} \qquad (4)$$

Further development of Eq. (4) using Eq. (2) make no difference with Eq. (3). From Eq. (3), one cannot obtain Minimal Cut-Set (MCS) because the term, S1, in RHS cannot be eliminated. Unless any other constraint is given, one cannot solve the Eq. (3) or (4).

In view of the initial state of a system, Eq. (3) can be solved by two ways depending on the initial conditions of S1 as follows:

(a) Supporting system 2 causes starting failure of system 1

If the supporting system 2 shall cause starting failure of system 1, it indicates that the system 1 is initially in the state of standby before its failure. If the system 1 is initially standby, system 2 cannot be supported by system 1. In terms of Boolean expression, the S1 in RHS of Eq. (3) should be treated as a universal event as follows

$$\begin{aligned} S_1 &= c_1 + a_{12}c_2 + a_{12}a_{21}.true \\ &= c_1 + a_{12}c_2 + a_{12}a_{21} \end{aligned} \qquad (5)$$

To investigate the reality of the cutest in Eq. (5), let's suppose a case that is described in Table 1 column 2. This case is a simple example of system initially in standby state. In this case, since the electric generator (EG) would be started when it receive its starting signal, the initial state of EG is in standby state. From the Eq. (5), the EG will not be operated with the following combinations of failure.

(1) {a mechanical failure of EG}

(2) {a failure of manual actuation, mechanical, a failure of signal generator}

(3) {a failure of manual actuation, a failure of internal electricity source}

(b) Supporting system 2 causes running failure of system 1:

If the supporting system 2 shall cause running failure of system 1, it indicates that the system 1 is initially in the state of running before its failure. As shown in Fig. 1, if the system 1 is initially in the running state, system 2 cannot be failed by the supporting failure of system 1. The S_1 in RHS of Eq. (3) should be treated as a null event. The Minimal Cut Set (MCS) can be determined by inserting the initial condition of system 1 into Eq. (3) as follows:

$$S_1 = c_1 + a_{12}c_2 + a_{12}a_{21}.false = c_1 + a_{12}c_2 \quad (6)$$

As an illustrative example, let's suppose the event set in column three of Table. 1. The EG will not be operated with the following combinations of failure as is shown in Eq. (6)

(1) {mechanical failure of EG (running failure)}

(2) {failure of internal cooling device, mechanical failure of external cooler}

Table 1 Example of circular logics

Event	Standby case	Running Case
S_1	Failure of electric generator	Failure of electric generator
c_1	Mechanical failure of electric generator(standby)	Mechanical failure of electric generator(running)
a_{12}	Failure of manual actuation of electrical generator	Failure of internal cooling device
S_2	Failure of signal generator for the operation of electric generator	Failure of external cooler for electric generator
c_2	Mechanical failure of signal generator	Mechanical failure of external cooler
a_{21}	Failure of internal electricity source	Failure of internal electricity source

2.2 Treatment of CL generated by among multiple supporting systems

The example treatment method of CL in Section 2.1 can be extended to complex CLs generated by multiple supporting systems. When a multiple system is interrelated with other systems, the Boolean equation of each system can be written as [4]

$$S_i = c_i + \sum_{j \neq i} a_{ij} S_j \quad (7)$$

The general solution of Eq. (7) is given as follows [8]

$$S_i = c_i + \sum_{\substack{j_1=1 \\ j_1 \neq i}}^{n} a_{ij_1} \left(c_{j_1} + \sum_{j_2 = i} a_{j_1 j_2} \delta_{j_2} + \sum_{\substack{j_2=1 \\ j_2 \neq i, j_1}}^{n} a_{j_1 j_2} \left(c_{j_2} + \sum_{j_3 = i, j_1} a_{j_2 j_3} \delta_{j_3} + \cdots \right. \right.$$
$$\left. \left. + \sum_{\substack{j_{n-1}=1 \\ j_{n-1} \neq i, j_1, \cdots j_{n-2}}}^{n} a_{j_{n-2} j_{n-1}} \left(c_{j_{n-1}} + \sum_{j_n = i, j_1, \cdots j_{n-2}} a_{j_{n-1} j_n} \delta_{j_n} \right) \cdots \right) \right) \quad (8)$$

Where

$$\delta_i a_{ij_1} a_{j_1 i} \cdots = S_i a_{ij_1} a_{j_1 i} \cdots, \begin{cases} \delta_i = false\ if\ the\ self\ recursion\ term\ has\ no\ event\ set \\ \delta_i = ture\ if\ the\ self\ recursion\ term\ has\ a\ event\ set \end{cases}$$

The terms containing δ in Eq. (8) represent the CLs generated in system Si. These terms can be treated depending on the initial condition of Si in each CL. As explained in the section 2.1, if the system, Si is not performing its function, δ will be treated as universal event/true while it can be treated as null event/false if the initial condition of Si is in performing its function.

When one want to find an initial condition in a complex sequence of a CL, it may be difficult to search the condition easily. The method of finding initial condition of Si will be discussed in section 3.

3. Conclusions

In the present paper, we suggest the criteria for the treatment of CL, an initial condition for a specific CL. It was discussed that there can be two initial conditions for a CL, a standby state and a running state of a system. When a system is in the state of standby, the top event of a system in a FT should be treated as a universal event in the sense of Boolean algebra. In a similar manner, when a system is in the state of running, the top event of a system in a FT should be treated as a null event.

REFERENCES

[1]Carlson DD. Interim reliability evaluation program procedure guide, NUREG/CR-2728, SAND82-1100, US Nuclear Regulatory Commission, Washington, DC, 1983

[2]Coles GA, Powers TB, Breaking the logical loop to complete the probabilistic risk assessment. Proceeding of PSA 89: International Topical Meeting probability, reliability, and safety assessment, Pennsylvania, USA 1989 pp. 1155-1160, 1989

[3] Yang J. E., Han S. H., Park J. H., Jin Y. H., Analytic Method to break logical loops automatically in PSA, Reliability Eng. & System Safety, Vol 56, pp. 101-105

[4] Lim H. G., Jang S. C., An analytic solution for a fault tree with CLs in which the systems are linearly interrelated, Reliability Eng. & System Safety, Volume 92, Issue 6, June 2007, Pages 804-807