



An exact method for solving logical loops in reliability analysis

Takeshi Matsuoka*

Utsunomiya University, 7-1-2 Yoto, Utsunomiya City, Tochigi 321-8585, Japan

ARTICLE INFO

Article history:

Received 4 November 2008

Received in revised form

11 January 2009

Accepted 27 January 2009

Keywords:

Logical loop

Boolean equation

Exact method

Reliability analysis

GO-FLOW

ABSTRACT

This paper presents an exact method for solving logical loops in reliability analysis. The systems that include logical loops are usually described by simultaneous Boolean equations. First, present a basic rule of solving simultaneous Boolean equations. Next, show the analysis procedures for three-component system with external supports. Third, more detailed discussions are given for the establishment of logical loop relation. Finally, take up two typical structures which include more than one logical loop. Their analysis results and corresponding GO-FLOW charts are given. The proposed analytical method is applicable to loop structures that can be described by simultaneous Boolean equations, and it is very useful in evaluating the reliability of complex engineering systems.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Many kinds of techniques are used for the design of high reliable and safe systems, for example redundant components, back-up function by standby component, principle of diversity, and so on. In nuclear power plants, a mainline system is often supported by multiple sub-systems, which provide electricity or cooling water or lubrication oil. These support systems are sometimes mutually supported and/or recursively supported by mainline system. In the typical design of nuclear power plants, the electricity of sub-systems is fed by mainline system [1]. But this system configuration leads to a problem of solving logical loops in reliability analysis. There is no assurance of properly evaluating the reliability of a system which contains loop configurations. So, it is not certain that the present designs are adequate or optimal for the efficient usage of resources. Up to now, many attempts have been made as breaking the logical loops at the points where the dependencies among the support system are relatively weak and developing new logic without loops [2]. The authors have also proposed a method to break the logical loop in the GO-FLOW analysis [3]. Another method was to break the logical loops automatically based on the proposed criteria [4]. Recently, a recursive method has been pointed out [5]. But general solution for a logical loop was not known. This paper presents an exact method for solving logical loops in reliability analysis by using Boolean algebra.

2. Basic idea to solve logical loop

First consider two-component system, in which components are mutually supported as shown in Fig. 1. The letters A and B indicate components. The A_{out} and B_{out} are component's outputs as electric power or cooling water. The A_{in} and B_{in} are inputs to component A and B , respectively. With the input A_{in} , that is, with the support of component B , component A works and produces output A_{out} .

Here the symbol, A_s , is used to denote the event (set) in which component A is sound state with the corresponding symbols. The symbol, A_{out} , denotes event (set) in which component A successfully produces output. Then, using the Boolean notation,

$$A_{out} = A_s A_{in}.$$

Based on the Boolean algebra $A_s A_s = A_s, \dots$, and the relations $A_{in} = B_{out}, \dots$, above equation is transformed as follows:

$$\begin{aligned} A_{out} &= A_s A_{in} = A_s B_{out} = A_s B_s B_{in} = A_s B_s A_{out} \\ &= A_s B_s A_s A_{in} = A_s B_s B_{out}. \end{aligned} \quad (1)$$

It seems that Eq. (1) suggests us a logical loop produces tautological relation between the A_{out} and B_{out} . But Eq. (1) gives us important information about the A_{out} , that is, $A_{out} = A_s B_s A_{out}$. This relation is equivalent to the next Boolean equation.

$$X = \alpha X. \quad (2)$$

Our task is to find out the unknown element X that satisfies Eq. (2). It is seen that $X = m\alpha$ is a solution of Eq. (2), where m is an arbitrary Boolean element. Then A_{out} is mathematically

* Tel./fax: +81 28 689 6053.

E-mail address: mats@cc.utsunomiya-u.ac.jp

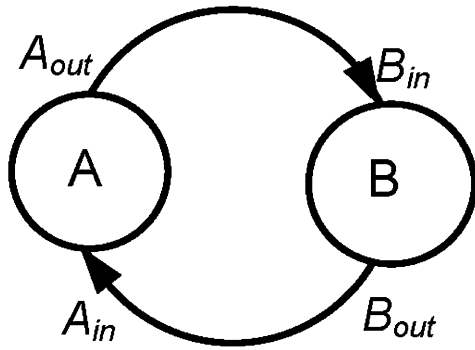


Fig. 1. Two components with mutual support.

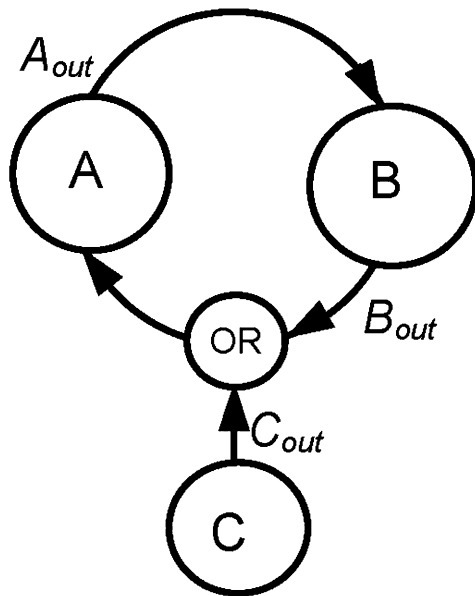


Fig. 2. Two components with additional component.

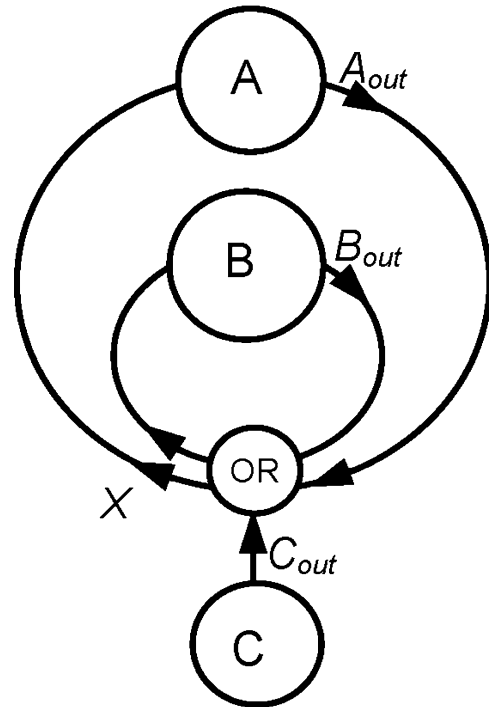


Fig. 3. Two components in parallel position.

determined as follows:

$$A_{out} = mA_sB_s. \quad (3)$$

If m is less than 1.0, the solution represents a situation that the two-component system shown in Fig. 1 is not used of its full ability. With the full operation of components A and B, the A_{out} becomes A_sB_s . A reliability of a system or a component is the maximum value of its available performance. Then the solution of Eq. (2) is $X = \alpha(m = 1)$, if a Boolean equation represents the relation among success events of components' states or outputs in engineering system.

Next, consider the condition in which an additional component C is connected to the two-component system as shown in Fig. 2. Then, A_{in} is expressed by the following equation:

$$A_{in} = B_{out} + C_{out} = B_sB_{in} + C_{out} = B_sA_{out} + C_{out} = B_sA_sA_{in} + C_{out}. \quad (4)$$

This is equivalent to the next equation.

$$X = \alpha X + \beta. \quad (5)$$

The solution of Eq. (5) is

$$X = m\alpha + \beta, \quad (6)$$

with an arbitrary Boolean element m .

Before component C is connected, $A_{in} = A_{out} = B_{in} = B_{out} = A_sB_s$. After the connection, A_{out} does not change because A_{out} represents

an event in which components A and B are in full operation. Then, $B_{out} = A_sB_s$.

Therefore, A_{in} becomes as follows:

$$A_{in} = B_{out} + C_{out} = A_sB_s + C_{out}. \quad (7)$$

This result means the solution of Eq. (5) becomes as follows:

$$X = \alpha X + \beta \Rightarrow X = \alpha + \beta. \quad (8)$$

Now, consider the system configuration in which components A and B are placed in parallel position as shown in Fig. 3. The Boolean relation becomes,

$$X = A_sX + B_sX + C_{out}. \quad (9)$$

This is equivalent to the next equation.

$$X = \alpha X + \beta X + \gamma. \quad (10)$$

The solution of Eq. (10) is,

$$X = m_1\alpha + m_2\beta + \gamma. \quad (11)$$

Arbitrary Boolean elements m_1 and m_2 are mutually independent. With the full operation of components A and B, $m_1 = m_2 = 1$ is also held.

Above rules are generally held if the Boolean equation represents the relation among success events of components' states or outputs in engineering system.

The next rules are easily derived as the special case of Eq. (8).

$$X = \alpha(X + \beta) \Rightarrow X = \alpha, \quad (12)$$

$$X = \alpha(\beta X + \gamma) \Rightarrow X = \alpha(\beta + \gamma). \quad (13)$$

Here, consider the complement of Eq. (12).

$$\bar{X} = \bar{\beta}\bar{X} + \bar{\alpha} \Rightarrow \bar{X} = \bar{\alpha}. \quad (14)$$

In Eq. (14), the Boolean elements represent events (sets) of components' failed states or no output. In this case, the solution of the form of Eq. (5) becomes $X = \beta$, that is $m = 0$ in Eq. (6).

The logic of a loop system is usually expressed by simultaneous Boolean equations, and the next equation is a general expression

for two unknown variables with complex supporting relations.

$$X = \alpha X + \beta Y + \gamma XY + \delta, \quad (15)$$

$$Y = \varepsilon X + \phi Y + \varphi XY + \eta. \quad (16)$$

If the unknown variables X and Y are expressed by known elements α, β, \dots , then the logical loop can be rigorously solved without approximation. The procedures are as follows. The Eq. (15) is transformed into the next equation,

$$X = (\alpha + \gamma Y)X + \beta Y + \delta. \quad (17)$$

By the rule of Eq. (8), X becomes,

$$X = \alpha + \gamma Y + \beta Y + \delta, \quad (18)$$

as α and β in Eq. (8) are not restricted to known elements.

Then this relation is substituted into Eq. (16).

$$Y = \varepsilon(\alpha + \beta Y + \gamma Y + \delta) + \phi Y + \varphi Y(\alpha + \beta Y + \gamma Y + \delta) + \eta \\ = \{\varepsilon(\beta + \gamma) + \phi + \varphi(\alpha + \beta + \gamma + \delta)\}Y + \varepsilon(\alpha + \delta) + \eta. \quad (19)$$

Again, by the aid of the rule of Eq. (8), Y becomes

$$Y = \varepsilon(\alpha + \beta + \gamma + \delta) + \phi + \varphi(\alpha + \beta + \gamma + \delta) + \eta. \quad (20)$$

This result is substituted into Eq. (18), and then X is determined as

$$X = \alpha + \beta(\varepsilon + \phi + \varphi + \eta) + \gamma(\varepsilon + \phi + \varphi + \eta) + \delta. \quad (21)$$

The unknown variables X and Y are determined, that is, they are expressed only by the combination of known elements α, β, \dots

Sometimes known elements α, β, \dots in Eq. (8) contain common sub-sets. In this case, we can use the rules of Eqs. (12) and (13) for the convenience of calculation. The unknown variables are determined and eliminated step-by-step with the aids of applicable rules of Eqs. (8), (12) and (13). By this procedure, an event (set) of component's successful output can be determined without any approximation for any element included in a logical loop system, if the system can be expressed by simultaneous Boolean equations.

3. Three-component system with external supports

3.1. Analytical solution

Now, consider three-component system as shown in Fig. 4. In this system, three main components A, B and C are connected by "OR" combinations. Element A_a does not rely on other component's support, and produces output for itself. It models an external support to this system like external electric power source or external water supply. There is an element A_b between A and B . It receives output of B and sends some quantity to A . The element A_b (and A_c) represents equipment or sub-system like an electric transformer, an energy converter, an amplifier and so on. Component A receives three supports and can work with one of them, that is, each element can supply enough quantity for the operation of component A . Components B and C are placed in the same condition as A in this system.

Then, with the Boolean notation,

$$A_{out} = A_s\{A_a + A_b B_{out} + A_c C_{out}\}. \quad (22)$$

The B_{out} and C_{out} are also expressed in the same form to Eq. (22). The A_{out}, B_{out} and C_{out} are unknown variables. Let replace them by X, Y, Z . Then, this three-component system is represented by the next simultaneous Boolean equations.

$$X = A_s\{A_a + A_b Y + A_c Z\}, \quad (23)$$

$$Y = B_s\{B_b + B_a X + B_c Z\}, \quad (24)$$

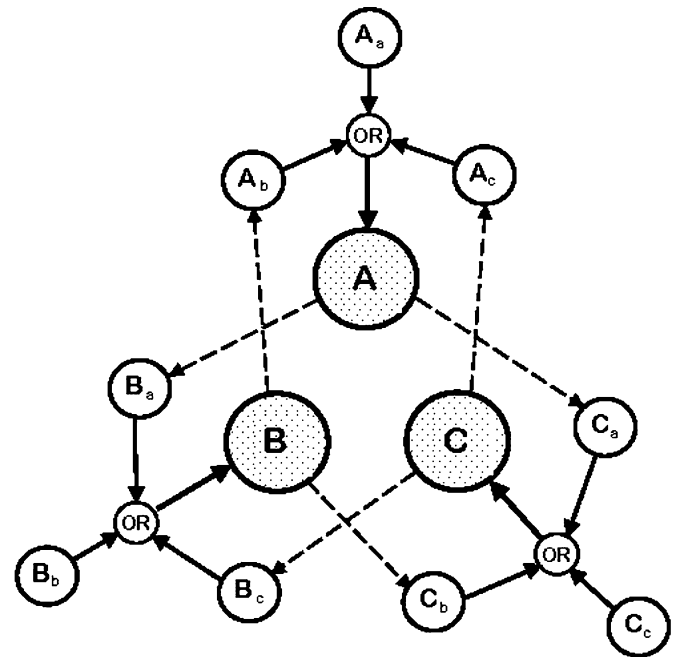


Fig. 4. Three components with external supports.

$$Z = C_s\{C_c + C_a X + C_b Y\}. \quad (25)$$

Our task is to express the unknown variables X, Y, Z by known elements A_a, B_a, \dots

Substitute Eq. (25) into Eqs. (23) and (24),

$$X = A_s\{A_a + A_b Y + A_c C_s C_c + A_c C_s C_a X + A_c C_s C_b Y\}, \quad (26)$$

$$Y = B_s\{B_b + B_a X + B_c C_s C_c + B_c C_s C_a X + B_c C_s C_b Y\}. \quad (27)$$

From Eq. (27), with the aid of Eq. (8), Y is determined as follows:

$$Y = B_s B_b + B_s B_c C_s C_b + B_s B_c C_s C_c + B_s (B_a + B_c C_s C_a) X. \quad (28)$$

Substitute Eq. (28) into Eq. (26), then

$$X = A_s\{B_s A_b B_a + C_s A_c C_a + B_s C_s A_b B_c C_a + B_s C_s A_c B_a C_b\}X \\ + A_s A_a + A_s C_s A_c C_c + A_s B_s A_b B_b + A_s B_s C_s A_b B_c C_b \\ + A_s B_s C_s A_b B_c C_c + A_s B_s C_s A_c B_b C_b + A_s B_s C_s A_c B_c C_b. \quad (29)$$

Finally, X is determined with the aid of Eq. (8).

$$X = A_s A_a + A_s B_s A_b B_b + A_s C_s A_c C_c + A_s B_s A_b B_a + A_s C_s A_c C_a \\ + A_s B_s C_s \{A_b B_c C_a + A_b B_c C_b + A_b B_c C_c + A_c B_a C_b \\ + A_c B_b C_b + A_c B_c C_b\}. \quad (30)$$

Then, Y is determined by substituting Eq. (30) into Eq. (28), and Z is obtained from Eq. (25). All the unknown variables A_{out}, B_{out} and C_{out} can be represented by known elements A_a, B_a, \dots . For any simultaneous Boolean equations, the same procedure is applicable, and the reliability (success probability) of component's successful output, X_{out} (unknown variable), can be determined.

3.2. Comparison with previous works

Many attempts have been made for solving the problem of logical loops. Some of these attempts are compared with the present method.

Coles and Powers [2] took up a fault tree of "Loss of Low Pressure Water system", in which the top event appears again at the bottom of fault tree branch. At that gate, he omitted the branch which contains the top event for the third time, and gave term "Mechanical failure of #1 pump" \times "Mechanical failure of #2

pump". This term is combined with "Failure of MPWS (Medium Pressure Water system)" by AND gate in this fault tree. The term "Mechanical failure of #1 pump" × "Mechanical failure of #2 pump" × "Failure of MPWS" disappears because the single event "Mechanical failure of #1 pump" is combined with this term by OR gate. The loop logic of this fault tree has been solved by the present method with the assumption that same electric power is used for #1 and #2 pumps. The answer is just to remove the term "Loss of LPWS" × "Failure of MPWS" at the bottom of fault tree. Coles & Powers accidentally reached the correct answer.

Vaurio [5] has presented the recursive method for finding the solution of simultaneous Boolean equations. In his method, zero input is given to unknown variables for the start. Eq. (1) in Ref. [5] has the same form to the Eqs. (23), (24) and (25) with $A_s = B_s = C_s = 1$. Vaurio gave an answer consist of 5 terms for this equation, but the Eq. (30), which consists of 11 terms, is obtained as the solution of this simultaneous Boolean equation. The reason of this difference is as follows. Vaurio considered failure events as Boolean elements. Start with zero inputs was equivalent to $m = 0$ as shown in Eq. (14). If success events (reliability) were considered, the recursive method had to be started with 1 (one) input. Vaurio's recursive method gave a correct solution for this simultaneous Boolean equations.

Lim and Jang [6] have also tried to solve circular logics. In their method, unknown variables are recursively substituted into original equation, and if an unknown variable appears again, it is replaced by δ_i . Finally, unknown variable is expressed only by known elements and δ_i . The $\delta_i = 1$ or $0(\Phi)$ is determined based on whether it is an event or meaningless. The Eq. (13) in Ref. [6] is the same form to Eqs. (23)–(25), and they gave an answer consists of 11 terms including δ_i for this equation. They also considered failure events as the elements of Boolean equation. So, correct answer (consist of 5 terms) had to be deduced by engineering judgments.

4. Establishment of logical loop relation

In the previous sections, it is assumed that components are independently started their function at the beginning and mutual supports and logical loop relations are suddenly established. In actual engineering systems, supporting relations are established step-by-step. So, consider more precisely the beginning of system operation.

If there is only one loop structure, whole plant system could be simplified by lumping the components or sub-systems together, which are placed at the same position form a topological viewpoint. Fig. 5 shows a generalized system configuration for a single loop structure. The U means upstream part against loop structure, which includes an external power source. The D is downstream from loop. The parts A and B are the elements of loop, and only A directly connects to D .

Now, consider the irreversible change of states and let denote $P(q_D)D_s(\lambda_D, t)$ as the event (set) of successful start and following sound state of D as the function of time. Where, q_D is success probability of sub-system D 's start on demand, λ_D is failure rate of sub-system D , and t is time duration in which D is operating.

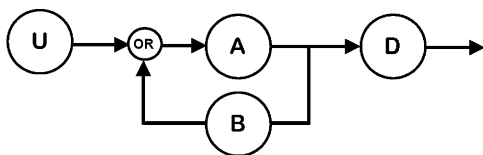


Fig. 5. Generalized system with one loop structure.

Suppose the following relation holds:

$$D_s(\lambda_D, t) = D_s(\lambda_D, t_0)D_s(\lambda_D, t - t_0). \tag{31}$$

At first, only the sub-system U is operating and the sub-system A takes t_1 hours for its start with the support of U 's output (power supply). Then B starts its operation with the support of A 's output at time t_2 ($t_2 > t_1$), and finally D starts at time t_3 ($t_3 > t_2$). The $P(q_A)$ has the following condition:

$$P(q_A) = 0(\phi); (t < t_1), \quad P(q_A) = const.; (t \geq t_1). \tag{32}$$

Just after the beginning of B 's operation,

$$\begin{aligned} A_{out}(t_2) &= P(q_A)A_s(\lambda_A, t_2 - t_1)P(q_U)U_s(\lambda_U, t_2), \\ B_{out}(t_2) &= P(q_B)B_s(\lambda_B, 0)P(q_A)A_s(\lambda_A, t_2 - t_1) \\ &\quad \times P(q_U)U_s(\lambda_U, t_2), \end{aligned} \tag{33}$$

where $A_{out}(t_2)$ denotes event (set) of sub-system A 's successful output at time t_2 . These equations are time dependent Boolean algebraic relations. After the start of B 's operation, logical loop relation is established. This is equivalent to connect the output of B to "OR" in Fig. 5, at time t_2 . But, just after the connection, there will not happen anything about the operational condition of sub-systems A and B , neither an additional start nor stop, because the $B_{out}(t_2)$ is a sub-set of $P(q_U)U_s(\lambda_U, t_2)$. An event $A_{out}(t_2)$ is expressed as

$$\begin{aligned} A_{out}(t_2) &= P(q_A)A_s(\lambda_A, t_2 - t_1)\{P(q_U)U_s(\lambda_U, t_2) + B_{out}(t_2)\} \\ &= P(q_A)A_s(\lambda_A, t_2 - t_1)\{P(q_U)U_s(\lambda_U, t_2) \\ &\quad + P(q_B)B_s(\lambda_B, 0)P(q_U)U_s(\lambda_U, t_2)\}. \end{aligned} \tag{34}$$

At time $t(t \geq t_2)$, the A_{out} becomes as follows:

$$A_{out}(t) = P(q_A)A_s(\lambda_A, t - t_1)\{P(q_U)U_s(\lambda_U, t) + B_{out}(t)\}. \tag{35}$$

$$\begin{aligned} A_{out}(t) &= P(q_A)A_s(\lambda_A, t_2 - t_1)A_s(\lambda_A, t - t_2) \\ &\quad \times \{P(q_U)U_s(\lambda_U, t_2)U_s(\lambda_U, t - t_2) + P(q_B)B_s(\lambda_B, t - t_2)A_{out}(t)\} \\ &= P(q_A)A_s(\lambda_A, t_2 - t_1)P(q_U)U_s(\lambda_U, t_2)A_s(\lambda_A, t \\ &\quad - t_2)U_s(\lambda_U, t - t_2) + P(q_A)A_s(\lambda_A, t_2 - t_1)A_s(\lambda_A, t \\ &\quad - t_2)P(q_B)B_s(\lambda_B, t - t_2)A_{out}(t). \end{aligned} \tag{36}$$

From Eq. (36), $A_{out}(t)$ is obtained as

$$\begin{aligned} A_{out}(t) &= P(q_A)A_s(\lambda_A, t_2 - t_1)P(q_U)U_s(\lambda_U, t_2)A_s(\lambda_A, t - t_2) \\ &\quad \times U_s(\lambda_U, t - t_2) + mP(q_A)A_s(\lambda_A, t_2 - t_1) \\ &\quad \times A_s(\lambda_A, t - t_2)P(q_B)B_s(\lambda_B, t - t_2). \end{aligned} \tag{37}$$

An arbitrary factor m in Eq. (6) appears again. In the limit of $t = t_2$, Eq. (37) equals to Eq. (34). Then, $m = P(q_U)U_s(\lambda_U, t_2)$. In this case, an arbitrary factor m is not unity. The part B is working with the support of A 's output and $A_{out}(t_2)$ is less than unity. Therefore, part B is not used in its maximum ability. $A_{out}(t)$ is determined as

$$\begin{aligned} A_{out}(t) &= P(q_U)U_s(\lambda_U, t_2)P(q_A)A_s(\lambda_A, t_2 - t_1) \\ &\quad \times \{A_s(\lambda_A, t - t_2)U_s(\lambda_U, t - t_2) + A_s(\lambda_A, t - t_2) \\ &\quad P(q_B)B_s(\lambda_B, t - t_2)\}; (t \geq t_2). \end{aligned} \tag{38}$$

The event (set) of total system's successful output becomes as follows:

$$\begin{aligned} D_{out}(t) &= P(q_U)U_s(\lambda_U, t_2)P(q_A)A_s(\lambda_A, t_2 - t_1)P(q_D) \\ &\quad \times D_s(\lambda_D, t - t_3)\{A_s(\lambda_A, t - t_2)U_s(\lambda_U, t - t_2) \\ &\quad + A_s(\lambda_A, t - t_2)P(q_B)B_s(\lambda_B, t - t_2)\}; (t \geq t_3). \end{aligned} \tag{39}$$

The result means that the $D_{out}(t)$ relies on both the upstream part (U) and logical loop parts (A and B) in addition to the D itself. If the part B is perfect ($P(q_B)B_s(\lambda_B, t) = 1$),

$$\begin{aligned} D_{out}(t) &= P(q_U)U_s(\lambda_U, t_2)P(q_A)A_s(\lambda_A, t - t_1)P(q_D) \\ &\quad D_s(\lambda_D, t - t_3); (t > t_3). \end{aligned} \tag{40}$$

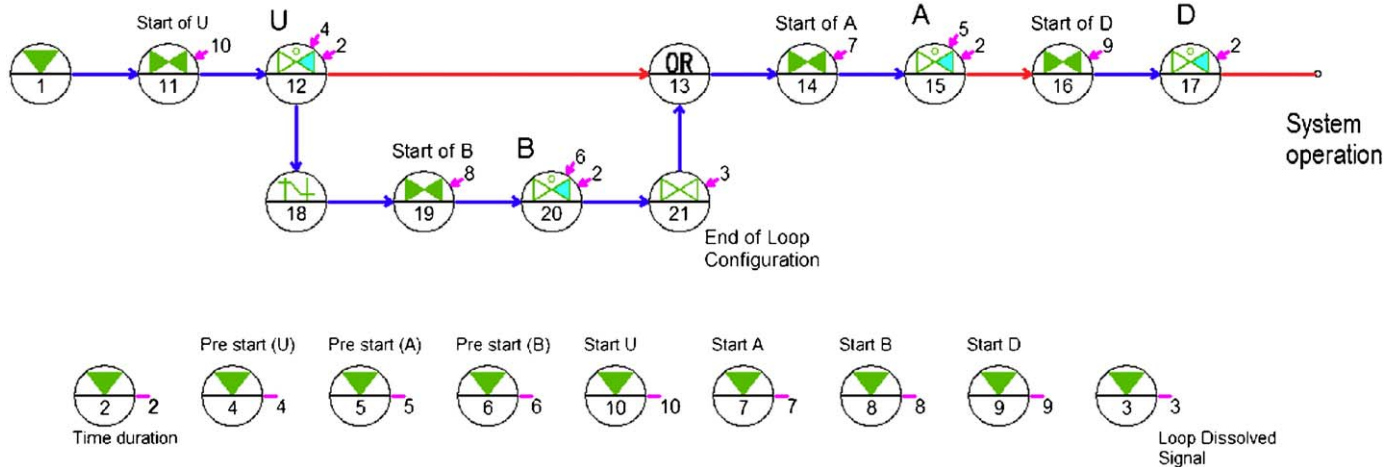


Fig. 6. GO-FLOW chart for the generalized system with one loop structure.

This equation suggests that the upstream part is only used for the start of A and B. On the other hand, if the reliability of B is very low,

$$D_{out}(t) \doteq P(q_U)U_s(\lambda_U, t)P(q_A)A_s(\lambda_A, t - t_1)P(q_D)D_s(\lambda_D, t - t_3); \quad (t > t_3). \quad (41)$$

The total system is mostly relied on U and D, that is, the effects of loop structure disappears.

Next, consider the case, the loop configuration is dissolved intentionally or by accident at time $t_4 (> t_3)$. The $D_{out}(t)$ becomes as follows:

$$D_{out}(t) = P(q_U)U_s(\lambda_U, t)P(q_A)A_s(\lambda_A, t - t_1)P(q_D)D_s(\lambda_D, t - t_3); \quad (t > t_4). \quad (42)$$

The contribution of $P(q_B)B_s(\lambda_B, t - t_2)$ is removed. The sub-system B in the return path of loop structure does not play any role after the loop is dissolved. Only the sub-systems on the main line in Fig. 5 should be considered.

These operational condition from $t = 0$ to t_4 is well modeled by the GO-FLOW [7] without any logical loop structure as shown in Fig. 6. In this GO-FLOW chart, operator No.1 is a signal generator and gives signal intensity of 1.0 at all the time points. This is the starting point of this system model. Sub-systems U, A, B, and D are modeled by type 37 operator; “failure of valve in open state”. Signal 2 gives time duration after the start of D; ($t > t_3$). Signals 4, 5, 6 give operating time duration before start of D, for U, A, B, respectively. With these signal connections, failure of each sub-system is properly modeled. Operator no.14 is a normally closed valve and it models the start of sub-system A. This valve is opened at time t_1 , by the command of signal 7. Start of D at time t_3 is modeled by the operator no.16 with the open command of signal 9. Effects of loop structure are expressed by the side route from operator nos.18 to 21. This structure is deduced from Eq. (39). Operator no.18 is phased mission operator and its function is to freeze the signal intensity after time t_2 . Out put signal intensity of no. 18 operator becomes $P(q_U)U_s(\lambda_U, t_2)$ at all the time points after t_2 . Operator no. 21 is a normally open valve, which is closed at time t_4 by the command of signal 3. This close action produces the system condition after the loop structure is dissolved.

Above GO-FLOW model is analyzed with the failure rates of $\lambda_U = 3 \times 10^{-3}/h$, $\lambda_A = 3 \times 10^{-4}/h$, $\lambda_B = 3 \times 10^{-4}/h$, $\lambda_D = 1 \times 10^{-3}/h$ and success probabilities of $q_U = q_A = q_B = q_D = 0.99$. Upstream part U's reliability and sub-system A's output are represented by the intensities of nos. 12 and 15 signals, respectively. The total system reliability is obtained by the intensity of no. 17 signal.

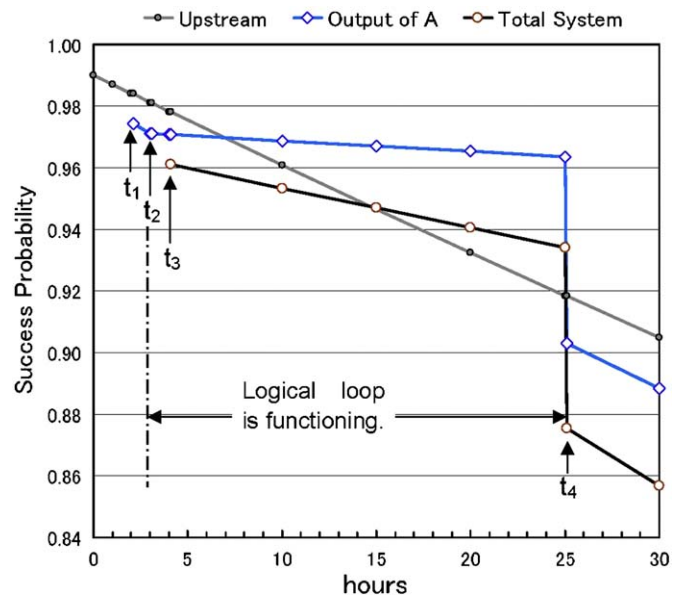


Fig. 7. Analysis result of one loop system.

Fig. 7 shows the analysis results of this loop system. Upstream part U's reliability constantly decreases with time. Sub-system A's output decreases slowly after the establishment of logical loop relation. The total system reliability also decreases slowly and becomes larger than U's reliability after 15 h by the effect of loop structure. After the dissolve of loop structure, sub-system A's output and total system reliability suddenly drop. They are mainly governed by the reliability of upstream U.

5. Multiple loop systems

Sometimes, large and complex systems have more than one loop structures. These loop structures may interact each other. Here, pick up two typical structures, analyze them by the present method and the results are expressed by the GO-FLOW charts without logical loop structure.

The first case is double loops with nested relation as shown in Fig. 8. The output of C should be also paid attention as “jump out from a loop”. In this section, it is assumed that operation of A is

started with the probability q_A at the beginning, and all other elements are started at time t_1 , and logical loop relations are established at this time t_1 . Element A is a special one in this system, because A can work without any support of other element. Event of element A 's successful output at time $t (> t_1)$ is $A_q A_0 A_1$, where A_q is event of element A 's successful start. Element A 's sound state is divided into two parts, before and after the start of other elements, that is, $A_s = A_0 A_1$. The events of elements E, D, C 's successful outputs are denoted by $X (= E_{out}), Y (= D_{out})$ and $Z (= C_{out})$, respectively. Then the Boolean relation becomes as follows:

$$X = E_q E_s Y, \tag{43}$$

$$Y = D_q D_s Z, \tag{44}$$

$$Z = \{(A_q A_0 A_1 + E'_q E'_s X) B_q B_s + D'_q D'_s Y\} C_q C_s. \tag{45}$$

Substitute the relations Eqs. (43) and (44) into Eq. (45),

$$\begin{aligned} Z &= \{(A_q A_0 A_1 + E'_q E'_s E_q E_s D_q D_s Z) B_q B_s + D'_q D'_s D_q D_s Z\} C_q C_s \\ &= A_q A_0 A_1 B_q B_s C_q C_s + C_q C_s D_q D_s E'_q E'_s E_q E_s B_q B_s Z \\ &\quad + C_q C_s D_q D_s D'_q D'_s Z \end{aligned} \tag{46}$$

Then Z becomes as follows by the rule of Eq. (11):

$$\begin{aligned} Z &= A_q A_0 A_1 B_q B_s C_q C_s + m_1 C_q C_s D_q D_s E'_q E'_s E_q E_s B_q B_s \\ &\quad + m_2 C_q C_s D_q D_s D'_q D'_s. \end{aligned} \tag{47}$$

At time t_1 , the Eq. (47) becomes next equation, because $A_1 = B_s = C_s = D_s = E_s = D'_s = E'_s = 1.0$.

$$Z = A_q A_0 B_q C_q + m_1 C_q D_q E'_q E_q B_q + m_2 C_q D_q D'_q. \tag{48}$$

Just after the establishment of loop structures, the event of C 's successful output (Z) is expressed in the following form from Fig. 8:

$$\begin{aligned} Z &= \{(A_q A_0 + A_q A_0 B_q C_q D_q E_q E'_q) B_q + A_q A_0 B_q C_q D_q D'_q\} C_q \\ &= A_q A_0 B_q C_q + C_q D_q E'_q E_q B_q A_q A_0 + C_q D_q D'_q B_q A_q A_0. \end{aligned} \tag{49}$$

Compare the Eqs. (48) and (49), then m_1 and m_2 are determined as follows:

$$m_1 = A_q A_0, \tag{50}$$

$$m_2 = B_q A_q A_0. \tag{51}$$

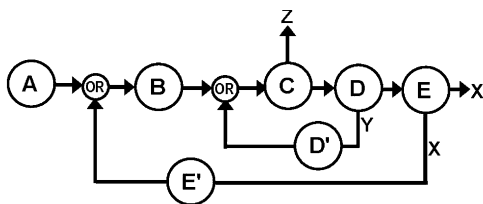


Fig. 8. Double loops with nested relation.

$X (= E_{out})$ and $Z (= C_{out})$ are obtained as

$$\begin{aligned} E_{out} &= A_q A_0 A_1 B_q B_s C_q C_s D_q D_s E_q E_s \\ &\quad + A_q A_0 B_q B_s C_q C_s D_q D_s E_q E'_s E'_q E'_s \\ &\quad + A_q A_0 B_q C_q C_s D_q D'_q D'_s E_q E_s, \end{aligned} \tag{52}$$

$$\begin{aligned} C_{out} &= A_q A_0 A_1 B_q B_s C_q C_s + A_q A_0 B_q B_s C_q C_s D_q D_s E_q E'_s E'_q E'_s \\ &\quad + A_q A_0 B_q C_q C_s D_q D'_q D'_s. \end{aligned} \tag{53}$$

In Eq. (52), the first term is the contribution from non-loop part. The second term is the contribution from the $B-C-D-E-E'$ loop structure and this loop is started by A 's output at time t_1 . The third term is the contribution from the $C-D-D'$ loop structure and the loop is started by B 's output at time t_1 . The result can be expressed by GO-FLOW charts as shown in Fig. 9. The phased mission operators are also used for the expression of the effects of logical loop relations. For the simple expression, operators which generate start signals for element's operations are omitted from the chart.

The second case is "Jump in from downstream". Fig. 10 expresses the situation that there are two "jump in" with crossed relation (G' and H') combined with one "jump out" to upstream (E'). The event of element F 's successful output becomes as follows:

$$\begin{aligned} F_{out} &= A_q A_0 A_1 B_q B_s C_q C_s D_q D_s E_q E_s F_q F_s \\ &\quad + A_q A_0 B_q B_s C_q C_s D_q D_s E_q E'_s E'_q E'_s F_q F_s \\ &\quad + A_q A_0 B_q C_q C_s D_q D_s E_q E_s F_q F'_s F'_q F'_s \\ &\quad + A_q A_0 B_q C_q D_q D_s E_q E_s F_q F_s G_q G_s G'_q G'_s \\ &\quad + A_q A_0 B_q C_q D_q E_q E_s F_q F_s G_q G_s H_q H_s H'_q H'_s. \end{aligned} \tag{54}$$

This result can also be expressed by the GO-FLOW chart as shown in Fig. 11. There are contributions from four loop structures, which are expressed by parallel inputs to No.13 "OR" gate in Fig. 11.

Above result seems complicated, but the expression by the GO-FLOW charts in Figs. 9 and 11 gives us a clear idea how to model loop relations. The contributions from loop structures are represented by side routes followed on phased mission operators, and they are connected to non-loop line through "OR" gate.

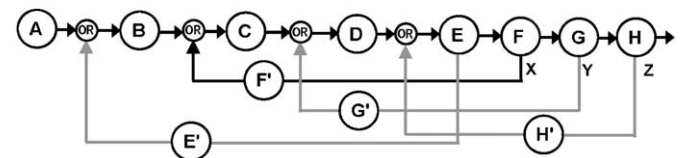


Fig. 10. Jump in from the downstream.

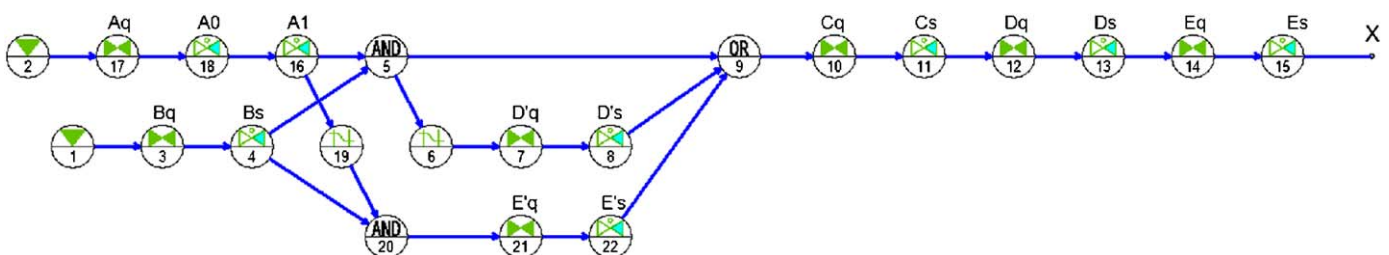


Fig. 9. GO-FLOW chart for the double loops with nested relation.

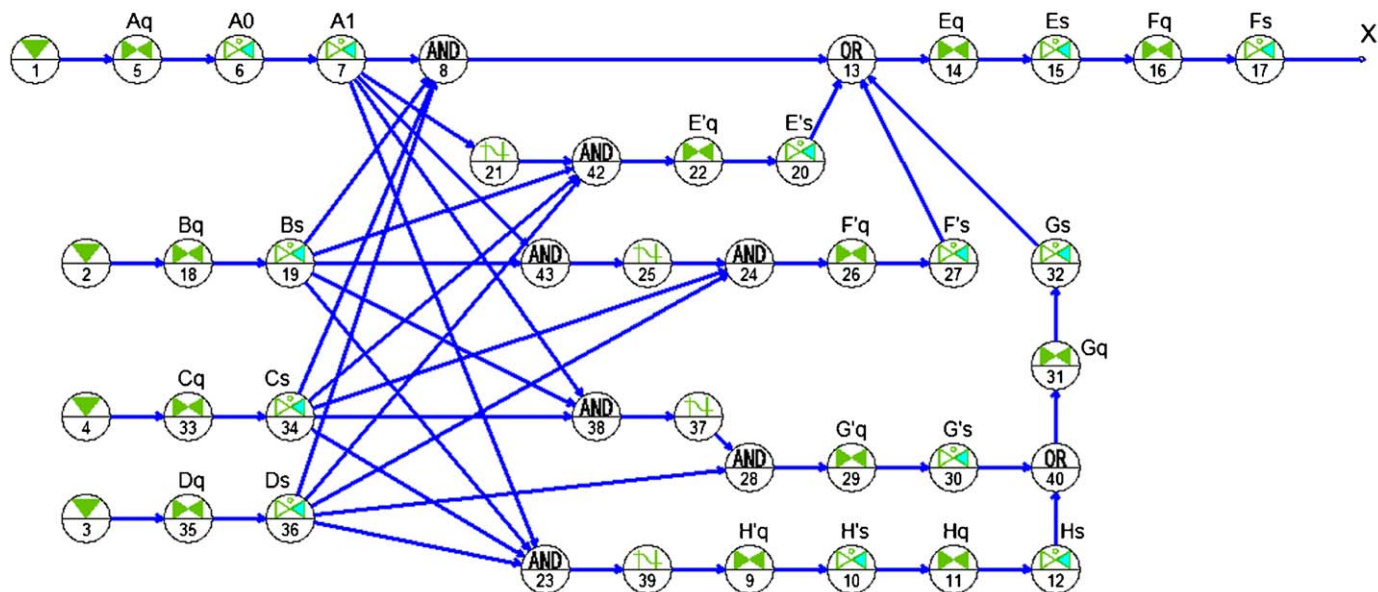


Fig. 11. GO-FLOW chart for the jump in from the downstream.

6. Conclusions

In the present paper, we presented an exact method for solving logical loops in reliability analysis by using Boolean algebra. The system that includes logical loop(s) is described by simultaneous Boolean equations. If this Boolean equations are analytically solved, logical loop(s) can be exactly solved in reliability analysis. The basic rule was given for Boolean algebraic equation. The procedures for solving logical loop were explained for three-component system with external supports.

In actual engineering systems, supporting and logical loop relations are usually established step-by-step. Then, detailed discussions were given for the establishment of logical loop and analysis result was shown for generalized one loop system.

By the present method, analyses were performed for two typical logical loop relations. Analysis results expressed by Boolean equations were shown with the corresponding GO-FLOW charts. The charts suggest us how to model logical loop relations without any loop structure.

The proposed exact method for solving logical loop is applicable to loop structures, which can be expressed by

simultaneous Boolean equations. The method is very useful in evaluating complex engineering systems, and also in effectively designing high reliable systems.

References

- [1] IAEA Safety standards series no. NS-G-1.8; Design of emergency power systems for nuclear power plants safety guide, IAEA, August 2004.
- [2] Coles GA, Powers TB. Breaking the logic loop to complete the probabilistic risk assessment. In: Proceeding of PSA'89: international topical meeting on probability, reliability and safety assessment, Pittsburgh, PA, April 2-7, 1989, American Nuclear Society, 1989. p. 1155-60.
- [3] Matsuoka T. A method to solve logical loops in the GO-FLOW methodology. In: Proceedings of PSAM-V; international conference on probabilistic safety assessment and management, Osaka, Japan, November 27-December 1, 2000, the International Association for PSAM, 2000. p. 1461-5.
- [4] Yang JE, Han SH, Park JH, Jin YH. Analytic method to break logical loops automatically in PSA. Reliab Eng Syst Saf 1997;56:101-5.
- [5] Vaurio JK. A recursive method for breaking complex logic loops in Boolean system models. Reliab Eng Syst Saf 2007;92:1473-5.
- [6] Lim HG, Jang SC. An analytic solution for a fault tree with circular logics in which the systems are linearly interrelated. Reliab Eng Syst Saf 2007;92: 804-7.
- [7] Matsuoka T, Kobayashi M. GO-FLOW: a new reliability analysis methodology. Nucl Sci Eng 1988;98:64-78.