

Acyclic nonlinear graph의 빠른 신뢰도 계산의 구현

이광원 이현규 성대현 윤승진
호서대학교 안전공학부

1. 서론

점점 과학의 고도화가 이루어지면서 산업 현장에서 사용하는 장비나 플랜트의 대형화, 복잡화가 이루어지고 있다. 이에 이러한 시스템들의 정량적 위험성 평가가 요구하기 시작했고, 이를 계산하기 위해서 ETA, FTA, CCA 등 여러 가지 방법들이 사용되고 있다. 그 중에서도 가장 보편적으로 많이 사용하는 방법으로 FTA(Fault Tree Analysis)를 손꼽을 수 있다. 관찰하는 시스템이 커지는 경우 FTA를 수행하기 위해서는 복잡한 계산 과정을 거치면서 많은 항의 계산을 하게 된다. 이러한 계산시간을 줄이기 위해서 많은 노력들이 있었으며 이중 하나는 FT를 graph(acyclic nonlinear graph)로 변형시킨 후 graph이론을 사용하여 신뢰도를 계산하는 것이다.

본 논문에서는 FT나 공통원인고장분석등에서와같이 하나의 부품이 여러번 나타나는 non-linear 형태의 graph에 있어서 빠르게 시스템의 신뢰도와 cutset들을 산출할 수 있는 알고리즘을 제시하고 이를 C-언어를 사용하여 구현하였다.

2. 이론적 배경

본 연구에서는 cycle을 포함치 않고 한 개의 선이 여러 곳에서 나타날 수 있는 acyclic nonlinear graph $G=(V,E)$ 를 관찰한다.

이 그래프는 절점들의 집합 V 와 선들의 집합 E 로 구성되며 절점은 완벽하고 선들은 각각의 신뢰도를 갖는 것으로 가정한다.

먼저 수학적 배경을 이해하기 위해 필요한 용어들을 알아보고 이 논문에서 사용된 수학적 배경을 설명하도록 하겠다.

2-1) 용어정의

i) Formation, Domination

관찰하는 어떤 acyclic nonlinear graph를 G 라 하고 임의의 부분 그래프를 G_a 라 할 때 G 의 선집합을 $E=\{e_1, e_2, \dots, e_b\}$ 이라하며 임의의 선의 부분집합 $E_a(\subseteq E)$ 는 G_a 의 선집합이라 하자.

선집합 E 에 포함되는 모든 선들을 원소로하는 임의의 집합의 family를 $M(G)=\{M_1, M_2, \dots, M_n\}$ 라 할 때 $M(G)$ 의 어떤 부분집합 $\underline{M}_i=\{M_i, M_j, \dots, M_l\}$ 가

$$\bigcup_{j=i}^l M_j = E_a \quad (2-1)$$

인 경우 G_a 의 formation이라하며 \underline{M}_i 에 포함된 M_j 의 수가 짝수일 경우 짝수 formation, 홀수인 경우 홀수 formation이라고 한다.

N_c 와 N_o 가 각각 짝수와 홀수 formation의 수를 나타낸다고 할 때 $M(G)$ 를 기초로 하는 부분그래프 G_a 의 domination인 $d(G_a, M(G))$ 는

$$d(G_a, M(G))=N_o-N_c \quad (2-2)$$

로 정의된다.

본 연구에서는 $M(G)$ 가 m -cutset들의 family $C(G)=\{C_1, C_2, \dots, C_m\}$ 인 경우를 관찰하게 되며 이 경우 식(2-2)는

$$d(G_a, C(G))=N_o-N_c \quad (2-3)$$

이 된다.

ii) Indegree, Outdegree, Incut, Outcut

그래프상에서 어떤 절점 V_i 로 들어오는 선의 수를 Indegree, 나가는 선의 수를 Outdegree라고 하며 또 절점 V_i 로 들어오는 선의 집합을 Incut(V_i) 또는 간단히 I_i , 절점 V_i 에서 나가는 선의 집합을 Outcut(V_i) 또는 간략히 O_i 라 표시한다.

2-2) 그래프 G의 신뢰도

지금까지는 일반적으로 어떤 시스템의 신뢰도나 고장도를 구하기 위하여서는 먼저 그 시스템의 m-pathset, 또는 m-cutset을 구하고 그것을 바탕으로 하여 Inclusion-Exclusion식(Poincaré식)에 대입하여 계산하게 된다. 이때 cutset의 수를 m개라 했을 때 일반식은

$$Q(G) = q[\bigcup_{i=1}^m A_i]$$

$$= \sum_{i=1}^m q(A_i) - \sum_{i=1}^m \sum_{j < i} q(A_i A_j) + \dots + (-1)^{m-1} q(A_1 A_2 \dots A_m) \quad (2-4)$$

with A_i ; i번째 cutset에 포함되는 모든 부품이 고장나는 사건
 $q(A_i)$; 사건 A_i 가 일어날 확률

로 표현되고 계산해야할 항의 수는 $2^m - 1$ 개가 된다.

보통의 경우 위식에서 계산되어지는 항들은 같은 선들을 포함하며 부호가 반대인 것들이 많이 출현하게 되며 서로 소거되어진다. domination의 정의에 의하여 위식은 다음과 같이 쓸 수 있다.

$$Q(G) = \sum_{G_A} d(G_A, C(G)) \cdot q(G_A) \quad (2-5)$$

with $q(G_A)$: G_A 에 포함되는 모든 선들이 고장날 확률

식 (2-5)는 항의 수가 최대 2^b 개 (즉 그래프 G가 포함하는 선들에 대한 모든 부분 집합)가 되며 보통의 경우 cutset의 수 m은 b 보다 훨씬 크므로 식(2-5)의 항수는 식(2-4)의 항수에 비하여 훨씬 적다고 할 수 있다.

이때 graph G에서 여러번 중복되어 나타나는 선들을 각기 다른 성분으로 인식하여 만든 linear graph를 G^L 로 표시한 경우 신뢰도 관계는 다음과 같음이 [1]에서 증명되었다.

$$Q(G) = \sum_{G_A^L} d(G_A^L, C(G^L)) \cdot q(G_A) \quad (2-6)$$

with $\left\{ \begin{array}{l} G_A^L ; \text{ linear graph } G^L \text{의 subgraph} \\ C(G^L) ; G^L \text{의 } m\text{-cutset의 family} \end{array} \right.$

식(2-6)를 사용하기 위하여는 cutset family $C(G)$ 를 기초로한 임의의 subgraph G_A^L 에 대한 domination값을 계산하여야 한다.

문헌[3]에서는 이들을 계산할 수 있는 몇 가지 식을 제시하고 있다. 즉

i) 그래프 G^L 에서 어떤 내부 절점들의 집합 A에서 Incut I_A^L 를 그리고 또 다른 내부 절점들의 집합 B(단 $A \cap B = \emptyset$)에서 Outcut O_B^L 를 제거할 때 얻어지는 subgraph $G - I_A^L - O_B^L$ 들만이 domination값이 0이 아니다.

ii) 임의의 subgraph G_A^L 이

$$G_A^L = G^L - I_{A_1}^L - O_{B_1}^L = G^L - I_{A_2}^L - O_{B_2}^L = \dots$$

로 표시되고 n을 그래프 G^L 의 내부 절점수라 할때

$$m_e : n - |A_i| - |B_i| \text{가 짝수인 } G^L - I_{A_i}^L - O_{B_i}^L \text{의 수}$$

$$m_o : n - |A_i| - |B_i| \text{가 홀수인 } G^L - I_{A_i}^L - O_{B_i}^L \text{의 수}$$

이라면 cutset family $C(G)$ 를 기초로한 G_A^L 의 domination값은 다음과 같다.

$$d(G_A^L, C(G^L)) = m_e - m_o \quad (2-7)$$

식(2-6)에서 domination값이 0인 subgraph들은 신뢰도 계산에 불필요하고 domination값이 0이 아닐 수 있는 subgraph들의 관찰이 중요하다. 이들의 family $SIO(G^L)$ 는 다음과 같이 정의되며 총수는 3^n 개이다.

$$SIO(G^L) = \{ G^L - I_A^L - O_B^L \mid \forall A \subseteq N, \forall B \subseteq \{N - A\} \} \quad (2-8)$$

그래프 G^L 를 O_{SN}^L 으로 표시할 수 있다. 그러면 위식은 다음과 같이 변형된다.

$$SIO(G^L) = \{ O_{SA}^L - I_B^L \mid \forall A \subseteq N, \forall B \subseteq A \} \quad (2-9)$$

식 (2-6)은 이제 식(2-7)과 (2-9)와 함께 다음과 같이 쓸 수 있다.

$$Q(G) = \sum_{\forall (O_{SA}^L - I_B^L) \in SIO(G^L)} (-1)^{|A| - |B|} \cdot q(O_{SA}^L - I_B^L) \quad (2-10)$$

위식은 $q(O_{SA} - I_B) = q(O_{SA} - I_A) \cdot q(I_B - O_{\bar{A}} - (O_{SA} - I_A))$ 와 함께 다음과 같이 변형된다.

$$Q(G) = \sum_{A \subseteq N} q(O_{SA} - I_A) \cdot \sum_{B \subseteq A} q(I_B - O_{\bar{A}} - (O_{SA} - I_A)) \cdot (-1)^{|B|}$$

단, $q(\emptyset) = 1$ (2-11)

만약 어떤 절점 $b \in B$ 에서 들어오는 모든 선이 A절점 이외의 곳에서 출발한 선들이라면 (즉 $I_b \subset O_{\bar{A}}$) 위식에서

$(I_B - O_{\bar{A}} - (O_{SA} - I_A)) = (I_{B-b} - O_{\bar{A}} - (O_{SA} - I_A))$ 가 되며 $q(I_B - O_{\bar{A}} - (O_{SA} - I_A))$ 와 $q(I_{B-b} - O_{\bar{A}} - (O_{SA} - I_A))$ 는 반대 부호를 가지므로 서로 상쇄되어진다. 만약 $A(G)$ 를 내부절점들의 부분집합 A들 중에서, A에 포함되는 모든 절점 a에 대하여 $I_a \not\subset O_{\bar{A}}$ 를 만족시키는 A들의 family 즉,

$$A(G) = \{ \forall A \subseteq N \mid \text{단 } \forall a \in A ; I_a \not\subset O_{\bar{A}} \}$$

(2-12)

이라면 위식은 다음과 같이 표현된다.

$$Q(G) = \sum_{A \in A(G)} q(O_{SA} - I_A) \cdot \sum_{B \subseteq A} q(I_B - O_{\bar{A}} - (O_{SA} - I_A)) \cdot (-1)^{|B|}$$

(2-13)

또는

$$Q(G) = \sum_{A \in A(G)} Q_A$$

(2-14)

with $Q_A = q(O_{SA} - I_A) \cdot \sum_{B \subseteq A} q(I_B - O_{\bar{A}} - (O_{SA} - I_A)) \cdot (-1)^{|B|}$

가 된다.

A에 포함되는 모든 절점a에 대하여 I_a^L 은 각각 disjoint이다. 그러나

nonlinear인 경우 I_i 들은 서로 공통으로 포함하는 선들이 존재하게 되며 모든 $a \in A$ 에 대하여 $I'_a = I_a - O_{\bar{A}} - (O_{SA} - I_A)$ 가

- i) disjoint인 a 의 집합을 A_d
- ii) 공통 선들을 포함하는 a 들의 집합을 A_o

라 한다면 식 (2-14)의 Q_A 는 $I'_B = I_B - O_{\bar{A}} - (O_{SA} - I_A)$ 와 함께

$$Q_A = q(O_{SA} - I_A) \cdot \prod_{a \in A_d} (1 - q(I'_a)) \cdot \sum_{B \subseteq A_o} q(I'_B) \cdot (-1)^{|B|} \quad (2-15)$$

가 된다.

이때 A_o 에 포함되는 모든 I'_a 에 대하여 $A_o = \{a_1, a_2, \dots, a_i, \dots\}$ 라 하면

$$\begin{aligned} \sum_{B \subseteq A_o} q(I'_B) \cdot (-1)^{|B|} &= 1 - q(I'_{a_1}) - q(I'_{a_2}) - \dots - q(I'_{a_i}) \\ &\quad - \sum_{B \subseteq (A_o - a_1)} q(I'_B) \cdot (-1)^{|B|} \\ &\quad - \sum_{B \subseteq (A_o - a_1 - a_2)} q(I'_B) \cdot (-1)^{|B|} \\ &\quad \dots \\ &\quad - q(I'_{a_i}) \end{aligned} \quad (2-16)$$

$$\text{with } I'_B = I_B - I'_{a_i}$$

로 쓸 수 있다. 각 항에서 다시 모든 $a \in (A_o - a_1)$ 에 대하여 I'_a 가 disjoint인지, 아닌지에 따라 반복할 수가 있게 된다.

● 예 제

관찰되는 그래프는 내부 절점이 4개이고 선이 8개인 그래프이다(Fig2-1. 참조). 이중선 c 와 선 e 는 2번씩 나타난다. 이 그래프에서 두 번째의 c 와 e 선을 대문자 C 와 E 로 표기하고 다른 성분으로 인식하면 Fig2-1.에 대응하는 acyclic linear graph를 얻게 된다. (Fig2-2. 참조) 이 그래프에서 식(2-12)에 정의된 $A(G)$ 는 $\{\emptyset\}, \{V_1\}, \{V_2\}, \{V_1, V_2\}, \{V_1, V_3\}, \{V_1, V_4\}, \{V_2, V_3\}, \{V_1, V_2, V_3\}, \{V_1, V_2, V_4\}, \{V_1, V_3, V_4\}, \{V_2, V_3, V_4\}, \{V_1, V_2, V_3, V_4\}$ 등 12개이며 이에 따른 $SIO(G^1)$ 와 $SIO(G)$ 를 Fig2-3.에 도시하였다.

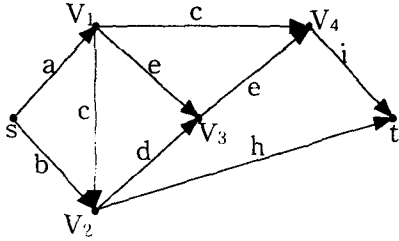


Fig2-1. acyclic nonlinear graph G

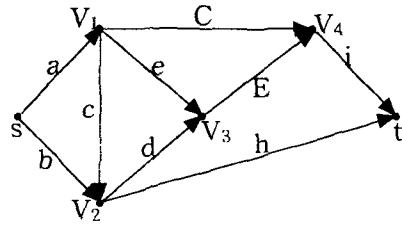


Fig2-2. Fig2-1에 대응하는 acyclic linear graph G^L

이 그래프의 고장도 $Q(G)$ 는 식(2-10)에 의하여 다음과 같이 쓰여진다.

$$\begin{aligned}
 Q(G) = & q(ab) \\
 & + q(bce) - q(abce) \\
 & + q(adh) - q(abdh) \\
 & + q(cdeh) - q(acdeh) - q(bcdeh) + q(abcdeh) \\
 & + q(bce) - q(abce) - q(bce) + q(abce) \\
 & + q(bcei) - q(abcei) - q(bcei) + q(abcei) \\
 & + q(aeh) - q(abe h) - q(adeh) + q(abdeh) \\
 & + (q(ceh) - q(aceh) - q(bceh) - q(cdeh) + q(abceh) + q(acdeh) \\
 & \quad + q(bcdeh) - q(abcdeh)) \\
 & + q(dehi) - q(adehi) - q(bcdehi) - q(dcehi) + q(abcdehi) + q(adcehi) \\
 & \quad + q(bcdehi) - q(abcdehi) \\
 & + q(bci) - q(abc i) - q(bcei) - q(bcei) + q(abcei) + q(abcei) \\
 & \quad + q(bcei) - q(abcei) \\
 & + q(ahi) - q(abhi) - q(adhi) - q(aehi) + q(abdhi) + q(abe hi) \\
 & \quad + q(adehi) - q(abdehi) \\
 & + q(hi) - q(bchi) + q(bcehi) - q(dehi) + q(cdehi) - q(cehi) \\
 & \quad - q(a) + q(abc) - q(abce) + q(ade) - q(acde) + q(ace)
 \end{aligned}$$

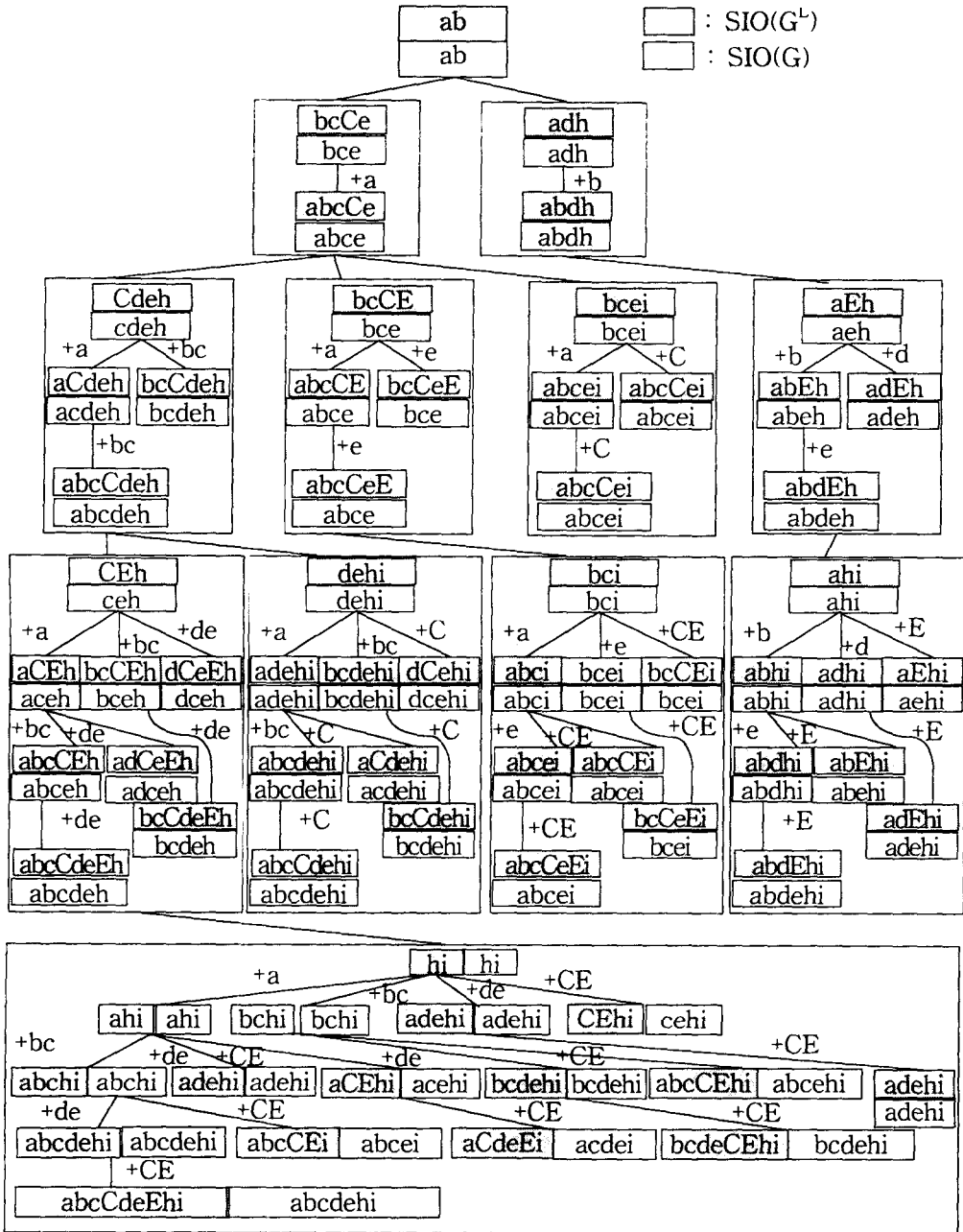


Fig2-3. tree형태로 표현된 SIO(G) 및 SIO(G^L)

다시 고장도 $Q(G)$ 는 식(2-13)에 의하여 다음과 같이 쓰여진다.

$$\begin{aligned}
 Q(G) = & q(ab) \\
 & + q(bce)(1 - q(a)) \\
 & + q(adh)(1 - q(b)) \\
 & + q(cdeh)(1 - q(a) - q(bc) + q(abc)) \\
 & + q(bce)(1 - q(a) - q(e) + q(ae)) \\
 & + q(bcei)(1 - q(a) - q(c) + q(ac)) \\
 & + q(aeh)(1 - q(b) - q(d) + q(bd)) \\
 & + q(ceh)(1 - q(a) - q(bc) - q(de) + q(ade) + q(bcde) - q(abcde)) \\
 & + q(dehi)(1 - q(a) - q(c) + q(ac)) \\
 & + q(bci)(1 - q(a) - q(e) + q(ae)) \\
 & + q(ahi)(-q(b) - q(d) + q(bd) + q(be) + q(de) - q(bde)) \\
 & + q(hi)(1 - q(a))(1 - q(bc) + q(bce) - q(de) + q(cde) - q(ce))
 \end{aligned}$$

위식은 다시 식(2-16)을 사용하여 다음과 같이 표현된다.

$$\begin{aligned}
 Q(G) = & q(ab) \\
 & + q(bce)(1 - q(a)) \\
 & + q(adh)(1 - q(b)) \\
 & + q(cdeh)(1 - q(a))(1 - q(bc)) \\
 & + q(bce)(1 - q(a))(1 - q(e)) \\
 & + q(bcei)(1 - q(a))(1 - q(c)) \\
 & + q(aeh)(1 - q(b))(1 - q(d)) \\
 & + q(ceh)(1 - q(a))(1 - q(bc))(1 - q(de)) \\
 & + q(dehi)(1 - q(a))(1 - q(c)) \\
 & + q(bci)(1 - q(a))(1 - q(e)) \\
 & + q(ahi)(1 - q(b))(1 - q(d))(1 - q(e)) \\
 & + q(hi)(1 - q(a))(1 - q(bc))(1 - q(e)) - q(de)(1 - q(c)) - q(ce)
 \end{aligned}$$

위식에서 계산되어지는 항수는 총 37개이며 m -cutset들은 $C(G) = \{\{a,b\}, \{b,c,e\}, \{a,d,h\}, \{a,e,h\}, \{c,e,h\}, \{b,c,i\}, \{h,i\}\}$ 로서 7개이고 Poincaré식에 의해 계산하려면 $2^n - 1 = 127$ 개를 계산하여야 한다.

3. 알고리즘

본 프로그램은 다음과 같이 main함수부, Recall1함수부, Recall2함수부의 3개의 주요함수로 구분되어지며 그 알고리즘은 다음과 같다.

Main program

Start

Step 1 : nonlinear graph를 linear graph로 바꾼다.

Step 2 : $A = \emptyset$, $Top^L = O^L_S$, $Q(G) = q(O_S)$

Step 3 : O^L_S 에 포함되는 모든 도착절점 i 에 대하여 Recall1(A, i, Top^L)호출

End

Recall1(A, i, Top^L)

Start

Step 4 : $A = A \cup i$, $Top^L = Top^L + O^L_i - I^L_i$, $Q = q(Top)$, $B = A$

Step 5 : B에 포함되는 모든 절점 i 에 대하여 $Son^L_i = I^L_i - O^L_B - Top^L$ 를 구한다.

Step 6 : B에 포함되는 모든 절점 i 에 대하여 Son_i 가 minimal이 아니면 제거
 $B = B - i$

Step 7 : B에 포함되는 모든 절점 i 에 대하여 Son_i 가 다른 Son 과 disjoint이면
 $Q = Q \cdot q(1 - q(Son_i))$
 $B = B - i$

Step 8 : 남은 절점집합 B가 없으면 goto Step 11.

남은 절점 집합 B가 $\{b_1, b_2, \dots, b_l\}$ 인 경우 모든 b_i 에 대하여

Step 9 : $Top^L_i = Son_i$
 $B_i = b_{i+1}, b_{i+2}, \dots, b_l$ $Q_i = q(Son_i)$
 $Recall2(B_i, Top^L_i, Q_i)$

Step 10 : $Q = Q(1 - Q_1 - Q_2 - \dots - Q_l)$

Step 11 : Top^L 에 포함되는 선들이 도착되는 내부절점들 k 에 대하여(단 $k \notin A$)
 $Recall1(A, k, Top^L)$

· STOP

Recall2(B, Top^L , Q)

start

Step 12 : B에 포함되는 모든 절점들 j에 대하여 $Son_j^L = I_j^L - O_j^L - Top^L$ 을 구한다.

$$Son = \{ Son_1, Son_2, Son_3, \dots, Son_k \}$$

Step 13 : B에 포함되는 모든 절점 i에 대하여 Son_i 가 minimal이 아니면 제거
 $B = B - i$

Step 14 : B에 포함되는 모든 절점 i에 대하여 Son_i 가 다른 Son 과 disjoint이면

$$Q = Q \cdot q(1 - q(Son_i))$$
$$B = B - i$$

Step 15 : 남은 절점 집합 B가 없으면 end

Step 16 : 남은 절점 집합 B가 $\{b_1, b_2, \dots, b_l\}$ 인 경우 모든 b_i 에 대하여

$$Top_i^L = Son_i$$

Step 17 : $B_i = b_{i+1}, b_{i+2}, \dots, b_l$ $Q_i = q(Son_i)$

$$Recall2(B_i, Top_i^L, Q_i)$$

Step 18 : $Q = Q(1 - Q_1 - Q_2 - \dots - Q_l)$

end

4. 결론

본연구에서는 FT의 graph 변환시 등에서 나타나는 cycle을 포함하지 않고 하나 이상의 선이 중복되어 나타나는 acyclic nonlinear graph에 대하여 관찰하여 보았다. Inclusion-Exclusion 식을 사용하던 기존방법보다 더욱 효율적인 것으로 밝혀진 domination 이론을 사용하여 불필요한 항들을 제거하고 꼭 계산에 필요한 항만을 구하여 계산하는 방법을 제시하고, 또 linear 일때와 유사하게, 묶어서 계산할수 있는 방법을 찾아내어 실제로 계산되는 항의 수를 더욱 감소시켰다. 이것을 예제를 통하여 보았을 때 계산하는 항이 현저히 감소함을 확인할수있었다. 그리고 이를 프로그램화하여 실용화 하였다.

앞으로는 서론부분에서 언급되었던 FTA나 공통원인고장분석방법등과 같이 하나의 사상이 중복되어 발생하는 위험성 평가의 방법의 경우에 본연구에서 제시

한 방법과의 접목을 통하여 현저한 계산시간의 향상을 가져올수 있으리라 예상되며, 학문적으로는 cycle을 포함한 cyclic nonlinear graph에서의 신뢰도 계산 방법 연구가 또 하나의 과제라 하겠다.

5. 참고문헌

- [1] 이일재, FT의 빠른 신뢰도 계산을 위한 연구, 호서대학교 산업안전공학과, 1997.
- [2] 이광원, 이일재, 강신재, Domination 이론에서의 새로운 식과 이의 신뢰성계산에 대한 적용, 한국 산업 안전 학회지, vol.11, No.1, 1996.
- [3] 이광원, Domination 이론을 이용한 acyclic digraph의 빠른 신뢰도 계산을 위한 연구, 한국 산업 안전 학회지, vol.11, No.1, 1996.
- [4] Kwang-won Rhie, Zur Domination und Zuverlässigkeit linearer Graphen aufgrund ihrer Minimalschnitte, KIssertation, Technische Universität in Berlin, July 1994
- [5] A. Satyanarayana, "A unified Formula for Analysis of Some network Reliability Problems," IEEE Trans. Reliability, vol R-31, No. 1, April 1982.