

삼상태 추론과 불 검증이 가능한 전문가 시스템에 관한 연구

손 동욱*, 박 영문, 윤 지호

서울 대학교 전기 공학과

A Study on the Expert System with
Three State Inference & Rule Verification

Son Dong-Wook, Park Young-Moon, Yoon Ji-Ho,

Dept. of Electrical Engineering, Seoul National University

Abstracts

Rules in expert system have meaning of assigning never-happen-minterms. Overall logical relations of variables can be achieved by making all prime implicants of never-happen-minterms. From prime implicants, two tables, which are necessary in the process of inference, are constructed. There are two inferencing modes. One executes inference only one variable which the user is interested in, and the other executes inference all variables simultaneously. Outputs of inference have not only 'true' or 'false', but also 'unknown', which is different from conventional expert system. In this paper, an efficient approach is presented, which can check logical inconsistency in knowledge base and contradiction between input facts and rules.

The methods in the paper may be available in the field of diagnosis and alarm processing.

(Contradiction) 을 발견할 수 없다. 만약 $A \rightarrow B, B \rightarrow C$ 라는 불이 있고 $A = 참, B = 거짓$ 이라는 데이터가 들어온다면 이는 불과 데이터간의 모순이 일어난 것이다.

또한 원래의 명제와 대우 명제는 논리적으로 동치이나 기존의 전문가 시스템에서는 추론시에 대우 명제를 동시에 고려하지 못한다. 예를 들어 $A \rightarrow B, B \rightarrow C$ 라는 불이 있고 C가 거짓이라는 데이터가 있는 경우를 생각해 보자. 논리적으로는 당연히 A, B가 거짓이라는 결론이 나오지만 기존의 전문가 시스템은 위의 불과 동치인 대우 명제 $\bar{A} \rightarrow \bar{B}, \bar{B} \rightarrow \bar{C}$ 를 불 베이스에 써주지 않는 한 이를 고려하지 못한다.

현재 전문가 시스템 구축시 불들을 검증하여 건전한 불 베이스를 구축하는 것이 중요한 관심사가 되고 있다. 기존의 전문가 시스템은 불간에 논리적 모순이 있을 경우에 이를 발견하지 못한다. 예를 들어서 $A \rightarrow B, A \rightarrow \bar{B}$ 라는 두 불이 공존한다면 이 두 불은 논리적으로 모순인데, 기존의 전문가 시스템에는 이들을 선별해서 사용자 (User) 에게 제시하여 사용자가 불을 수정할 수 있도록 하는 기능이 없다.

I. 서론

1. 기존의 전문가 시스템의 문제점

기존의 이전 논리를 다루는 전문가 시스템의 각 변수들은 참 (True)과 거짓(False)의 두 논리값만 가진다. 따라서 변수의 값을 모르는 상태 (Unknown) 과 거짓을 구분하지 않고 있다. $A \rightarrow B$ 라는 불을 생각하면 B의 값은 다음과 같다.

입력 데이터 (A)	출력 데이터 (B)	
	기존전문가 시스템	본 논문
false	false	unknown
true	true	true

A가 참일 경우 B가 거짓으로 되는 것은 불에 의해 금지되고 B는 참으로 고정된다. 그러나 A가 거짓일 경우 B는 참이 될 수도 있고 거짓이 될 수도 있다. 따라서 이러한 경우 (A=거짓) 에는 B는 미정 (Unknown) 으로 하는 것이 타당하다.

또한 기존의 전문가 시스템은 데이터와 불 간의 모순

II. 본론

1. 불과 금지 민텀 (Never-Happen-Minterm) 과의 관계

기존의 전문가 시스템에서는 추론 (Inference) 시에 불의 조건부 (Antecedent) 와 일치하는 데이터가 존재하는가를 탐색 (Search) 해서 이것이 데이터 베이스에 존재하면 불의 결론부 (Consequent) 를 데이터로 도출해낸다. 본 논문에서는 불의 의미를 다음과 같이 생각하였다. 아래에서, 기호는 논리합을, \cdot 기호는 논리곱을 나타낸다고 약속하자.

정의 1. 불은 금지 민텀 (Never-Happen-Minterm) 을 지정하는 역할을 한다. $A \rightarrow B$ 라는 불은 $\bar{A} \cdot B = 1$ (True)와 논리적으로 같은 의미이다. $\bar{A} \cdot B = 1$ 의 양변을 부정 (negation) 하면 $A \cdot \bar{B} = 0$ (False) 가 되고 이것은 민텀 (minterm) $A \cdot \bar{B}$ 가 금지됨을 의미한다. 따라서 불들은 일어날 수 없는 금지 민텀들을 지정하는 역할을 하며, 금지 민텀들은 불과 같은 의미를 내포하고 있다.

2. 금지 민텀과 이들의 프라임 임플리칸트 (Prime Implicant) 와의 관계

개개의 불은 불에 나타난 변수들만의 지역적(Local) 논리 관계를 명시하는 것이다. 불에 의해 생성된 각각의 금지 민텀들도 정리 1.에 의해서 이와 동일한 의미를 가진다. 본 논문에서는 이러한 지역적인 논리 관계에서 전체적(Global) 인 논리 관계를 찾아낸다.

정리2. 금지 민텀에서 이들의 프라임 임플리칸트(Prime Implicant)를 구하는 것은 모든 변수들간에 존재하는 논리적 관계를 밝히는 것이다.

예를 들어 $A \rightarrow B$, $B \rightarrow C$ 라는 두 불에서 금지 민텀은 $A \cdot \bar{B}$, $B \cdot \bar{C}$ 이고 이들을 카르노 맵 상에 나타내면 다음 그림과 같다.

	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
\bar{C}		○	○	○
C			○	○

원래의 금지 민텀 $A \cdot \bar{B}$, $B \cdot \bar{C}$ 에서 $A \cdot \bar{C}$ 라는 프라임 임플리칸트(점선으로 묶여진 프라임 임플리칸트)를 만들 수 있는데, 이 과정은 주어진 불베이스에 없었던 A와 C간의 논리적 관계를 찾는 것이다. 실제로 $A \cdot \bar{C}$ 라는 금지 민텀을 부정하면 $\bar{A} + C$ 가 되고, 이것은 $A \rightarrow C$ 라는 불로 생각할 수 있다.

모든 프라임 임플리칸트를 구하여 변수들간의 논리적 관계를 밝힌뒤, 이 프라임 임플리칸트 정보를 이용하여 추론에 필요한 테이블을 다음과 같이 만들 수 있다. 위의 예에서 구한 프라임 임플리칸트는 $A \cdot \bar{B}$, $B \cdot \bar{C}$, $A \cdot \bar{C}$ 이고 이들은 금지 민텀에서 구해진 프라임 임플리칸트이므로 이들의 논리합은 다음 식과 같이 항상 거짓이다.

$$A \cdot \bar{B} + B \cdot \bar{C} + A \cdot \bar{C} = 0 \quad (1)$$

식(1)의 양변을 부정하면 다음과 같은 항진식이 나온다.

$$(\bar{A} + B) \cdot (\bar{B} + C) \cdot (\bar{A} + C) = 1 \quad (2)$$

윗 식은 항진식이므로 각각의 괄호 안은 전부 참이 되어야 하는데 여기서 각 문자가 참 또는 거짓으로 결정되는 조건들을 찾을 수 있다. 예를 들어 식(2)에서 B가 참이 되기 위한 조건은 다음과 같이 찾을 수 있다. A가 참이라면 식(2)의 첫번째 괄호 안이 1이 되기 위해 B가 참으로 고정된다. 그리고 B가 참이라는 데이터가 직접 존재해도 된다. 마찬가지로 B가 참 (즉 B가 거짓)이 되기 위한 조건은 B가 거짓이라는 데이터가 직접 있거나 C가 거짓이라는 조건이 있으면 두번째 괄호안을 1로 만들기 위해 B는 거짓으로 된다.

위의 예에서 각 변수가 참 또는 거짓으로 고정되기 위한 조건들을 테이블로 구성한 것을 보이면 다음과 같다. 테이블의 오른쪽은 왼쪽에 나타난 변수값을 결정할 수 있는 데이터들의 집합이다. 이 테이블은 사용자가 관심있는 변수의 값에 대해 질문할때 이에 대답하기 위해 쓰이므로 Answer Table이라 부르기로 한다.

Answer Table은 프라임 임플리칸트에서 기계적으로 쉽게 구성할 수 있다. A가 참임을 증명하기 위해서는 프라임 임플리칸트들 중에서 \bar{A} 가 포함된 것들을 찾아 \bar{A} 를 제외 한 나머지항들을 +로 연결하면 된다.

A	A
\bar{A}	$\bar{A} + \bar{B} + \bar{C}$
B	A + B
\bar{B}	$\bar{B} + \bar{C}$
C	A + B + C
\bar{C}	\bar{C}

3. 불 간의 논리적 모순을 발견하는 방법

불의 실용적 의미가 맞는가 틀리는가를 떠나서, 그들이 논리적으로 상호 모순이 없는가를 밝히는 것은 다음과 같은 방법으로 할 수 있다.

정리3. 불간에 논리적 모순이 있으면 불이 데이터화된다.

$A \rightarrow B$, $A \rightarrow \bar{B}$ 라는 불이 있을 경우 각각의 금지 민텀은 $A \cdot \bar{B}$, $A \cdot B$ 이고 이들의 프라임 임플리칸트는 A (=거짓)이다. 이는 불 간에 논리적 모순이 있을 때, 두 불이 A는 거짓이라는 데이터로 바뀔을 의미한다.

정리3을 이용하여 프라임 임플리칸트를 만들때 데이터화 되는 불이 있는가를 검사함으로써, 불간의 모순을 찾아낼 수 있고 이를 사용자에게 제시하여 수정하도록 요구한다.

4. 테이블을 이용한 추론

위와 같은 방법으로 구성된 Answer Table에서 추론은 다음과 같은 방법으로 이루어진다. 먼저 입력된 데이터를 다음과 같이 코딩한다.

	A	\bar{A}
참	1	0
거짓	0	1
미정	0	0

이렇게 코딩된 값을 Answer Table의 오른쪽에 대입하고 논리값을 계산한다. 만약 결과가 $A = 1$, $\bar{A} = 0$ 이라면 A가 참임을 증명하는 데이터는 존재하나 A가 거짓임을 증명할 수 있는 데이터는 존재하지 않으므로 결과적으로 A는 참이 된다. 반대로 결과가 $A = 0$, $\bar{A} = 1$ 이면 A는 거짓이다.

그런데 불과 데이터간의 모순이 존재하는 경우 (예를 들자면 $A \rightarrow B$, $B \rightarrow C$ 라는 불이 있을때, A = 참, C = 거짓이라는 데이터가 들어오는 경우) 는 결과가 $A = 1$, $\bar{A} = 1$ 이 되어 모순이 일어났다는 것을 알 수 있다.

또한 데이터베이스에 A와 A가 참임을 증명할 수 있는 데이터들이 존재하지 않는다면 결과는 $A = 0$, $\bar{A} = 0$ 가 되며 A가 참인지 거짓인지 알 수 없으므로 이때 A는 미정(Unknown)이 된다.

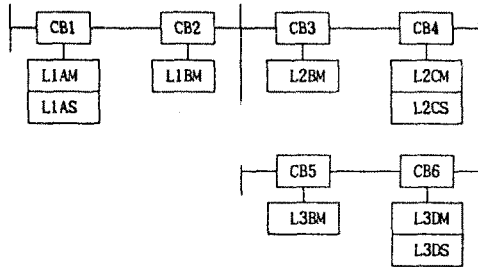
위와 같이 본 논문에서는 각 변수의 상태를 참, 거짓, 미정, 모순 등으로 구분하여 출력한다. 또한 사용자(User)에게 추론의 근거로써 테이블을 제시하여, 추론 과정을 설명할 수 있다.

위에서 제시한 Answer Table은 특정 변수의 값을 추론할 때 편리하나, 주어진 데이터의 집합에 의해서 모든 변수의 값을 동시에 추론하기 위해서는 또다른 형태의 테이블 (이를 Inference Table로 부르기로 한다.) 을 구성하는 것이

편리하다. 위의 Answer Table에서 A가 참이라는 데이터가 있으면 A, B, C가 동시에 참이 된다는 사실을 알 수 있다. Inference Table의 왼편에는 데이터를 기입하고 오른편에는 이러한 데이터에 의해서 값이 결정되는 변수와 그 변수 값을 기입한다. 추론시에는 데이터 베이스의 데이터가 Inference Table의 왼편과 매칭이 되면 테이블의 오른편에 기재된 변수들과 그 변수값을 출력하면 된다. 두가지 추론 모드에서 쓰이는 테이블들은 한번만 만들어 확일로 저장한 뒤 추론시에는 이 테이블을 이용한다.

표. 사례 연구

본 논문에서 제시한 방법을 다음과 같은 3-Line 1-bus 계통의 사고 구간 판정 문제에 적용시켜 보았다.



L1, L2, L3는 사고 구간을 나타내는 변수이고, CB는 회로차단기 (Circuit Breaker) 를 나타내며, LIAs, L2Cs, L3Ds는 후비 보호 릴레이이고, 나머지는 후보호 릴레이들이다. 이 계통에 대한 플베이스는 사고 구간과 릴레이, 릴레이와 회로 차단기 간의 관계를 나타내는 물들로 구성하였다.

고장구간 판단문제에 있어 각 변수들은 3종류로 구분할 수 있다. 첫째는 사용자가 그 값을 알고자 하는 변수 (Variable to be Verified) 이고, 두번째는 알람이 장치되어 데이터 값을 측정할 수 있는 변수 (Measurable Variable)이고, 세번째는 알람도 없고 사용자가 그 값을 알 필요도 없는 변수 (Hidden Variable) 이다. 이러한 사실을 고려하여 추론에 필요한 테이블을 만들때 사용자가 값을 알고자하는 변수에 대해서만 테이블을 만들고, 그 내용은 측정가능한 변수만으로 작성하면 테이블의 크기도 작아지고 추론도 효율적으로 할 수 있다.

프로그램 실행 예를 보이면 다음과 같다.

```
< INPUT DATA >
CB1 : TRUE      CB2 : TRUE
< RESULTS >
L1 IS TRUE.
L2 IS UNKNOWN.
L3 IS UNKNOWN.
CB1 IS TRUE.
CB2 IS TRUE.
CB3 IS UNKNOWN.
CB4 IS UNKNOWN.
CB5 IS UNKNOWN.
CB6 IS UNKNOWN.
LIAM IS UNKNOWN.
LIAS IS UNKNOWN.
L1BM IS TRUE.
L2BM IS UNKNOWN.
```

```
L2CM IS UNKNOWN.
L2CS IS UNKNOWN.
L3BM IS UNKNOWN.
L3DM IS UNKNOWN.
L3DS IS UNKNOWN.
```

위의 예는 CB1과 CB2가 트립되었다는 데이터만 입력되었을 때 inference table에 의해 모든 변수를 추론한 결과이다. 출력 결과로부터 L1 구간에 사고가 발생하였다는 사실을 알 수 있다.

```
< INPUT DATA >
LIAM : T      L2CS : T      L3DS : T
< RESULTS >
L1 IS TRUE.
L2 IS UNKNOWN.
L3 IS UNKNOWN.
CB1 IS TRUE.
CB2 IS FALSE.
CB3 IS UNKNOWN.
CB4 IS UNKNOWN.
CB5 IS TRUE.
CB6 IS TRUE.
LIAM IS TRUE.
LIAS IS UNKNOWN.
L1BM IS FALSE.
L2BM IS UNKNOWN.
L2CM IS UNKNOWN.
L2CS IS TRUE.
L3BM IS UNKNOWN.
L3DM IS UNKNOWN.
L3DS IS TRUE.
```

위의 예는 LIAM, L2CS, L3DS 릴레이가 동작하고 L1BM이 동작하지 않았다는 데이터만 입력되었을 때의 결과이다. 이때는 L1 구간에서 사고가 난 상태에서 L1BM 릴레이가 동작하지 않아서 L2CS와 L3DS 릴레이가 후비보호로 동작한 경우이다.

```
< INPUT DATA >
LIAM : T      L1BM : T      CB1 : F
< RESULTS >
L1 IS CONTRADICTION BETWEEN RULE & FACT.
L2 IS UNKNOWN.
L3 IS UNKNOWN.
CB1 IS CONTRADICTION BETWEEN RULE & FACT.
CB2 IS CONTRADICTION BETWEEN RULE & FACT.
CB3 IS UNKNOWN.
CB4 IS UNKNOWN.
CB5 IS UNKNOWN.
CB6 IS UNKNOWN.
LIAM IS CONTRADICTION BETWEEN RULE & FACT.
LIAS IS UNKNOWN.
L1BM IS CONTRADICTION BETWEEN RULE & FACT.
L2BM IS UNKNOWN.
L2CM IS UNKNOWN.
L2CS IS UNKNOWN.
L3BM IS UNKNOWN.
L3DM IS UNKNOWN.
L3DS IS UNKNOWN.
```

위의 예는 LIAM, LIBM은 동작하였는데 CBI은 동작하지 않았다는 데이터가 입력된 경우이다. LIAM, LIBM이 동작하였다는 정보에서 L1구간에서 사고가 난 것을 알 수 있는데, CBI브레이커가 동작하지 않았다는 데이터가 있으므로 이는 모순이다.

IV. 결론

본 논문은 기존 전문가 시스템의 문제점 중에서 거짓과 미정의 혼용 문제, 틀과 데이터간의 모순이 있을때 이를 검출하는 문제, 틀간의 논리적인 모순이 있을때 이를 발견하는 문제등을 명제논리에 한정하여 해결하였다.

그리고 테이블을 이용하여, 사용자가 관심있는 특정 변수의 값을 추론하거나 모든 변수의 값을 동시에 추론하는 두가지 추론 모드를 제시하였다.

이러한 방법은 특히 진단 문제, 알람 프로세싱등에 이용될 수 있을 것이다.

v. 참고 문헌

1. B.G.Buchanan, E.H.Shortliffe, " Rule - based Expert Systems ", Addison-Wesley, 1984.
2. D.D.Volfgan,T.J.dear,C.S.Galbraith," Expert Systems for the Technical professional", John Willey & Sons, 1987.
3. F.Hayes-Roth, D.A.Waterman, D.B.Lenat," Building Expert Systems ", Addison-Wesley, 1983.
4. Keinosuke Matsumoto, Toshiaki Sakaguchi,"An Approach to the Dynamic Verification of Knowledge-based Systems ", 2'nd simposium on Expert Systems Application to Power Systems,1989
5. J.Tepandi," Verification ,Testing, and Validation of Rule-based Expert Systems ", 1989
6. H.Sasaki, K.Kawahara," A Novel Scheme for Validation and Verification of Rule-bases in Expert Systems "
7. G Karnaugh, " The Map Method for Synthesis of Combinational Logic Circuits ", IEEE trnas. Commun.,v72, pp, 593-599,1953
8. N.J.Nilsson, "Principles of Artificial Intelligence" , Springer-Verlag, 1980.
9. R. davis, d. B. Lenat , " Knowledge-Based Systems Intelligence ",McGraw-Hill, 1982.
10. S. M. Weiss, C. A. Kulikowski, " A Practical Guide the designing Expert Systems ", Rowman & Allanheld,1984.
11. C.L. Chang, "Introduction to Artificial techniques ", JMA Press, INC., 1985.
12. M. Canegnen , D.H.D.Warren, " Logic programming and Applications ", Ablex Publishing Corporation, 1986.
13. Y. Sekine ," Expert system for fault section estimation of power system using time sequence information ", 3'rd symposium on expert systems application to power systems , 1991.
14. 황 회용, " 디지털 설계 ", 동일 출판사 , 1985.