

논리회로의 고장진단을 위한 퍼지 테스트생성 기법

조재희, 강성수, 김용기
국립경상대학교 컴퓨터학과

Fuzzy Test Generation for Fault Detection in Logic Circuits.

Jae-Hee Cho, Seong-Soo Kang, Yong-Gi Kim
Dept. of Computer Science, Kyungsang National University, Chinju, Korea

요 약

고집도 집적회로(VLSI)의 설계 과정에 있어 테스트(test)는 매우 중요한 과정으로서, 회로내의 결함(fault)을 찾기 위해 일련의 입력값을 넣어 그 출력값으로 고장 여부를 판단한다. 회로의 테스트를 위하여 사용되는 일련의 입력값을 테스트패턴(test pattern)이라하며 최고 2^n 개의 테스트패턴이 생성될 수 있다. 그러므로 얼마나 작은 테스트패턴을 사용하여 회로의 결함 여부를 판단하느냐가 주된 관점이 된다. 기존의 테스트 패턴 생성 알고리즘인 휴리스틱(heuristic) 조건에서 가장 큰 문제점은 빈번히 발생하는 백트랙(backtrack)과 이로 인한 시간과 기억장소의 낭비이다. 본 논문에서는 이러한 문제점을 보완하기 위해 퍼지 기법을 이용한 새로운 알고리즘을 제안한다. 제안된 기법에서는 고장신호 전파과정에서 여러개의 전파경로가 존재할 때, 가장 효율적인 경로를 선택하는 단계에서 퍼지 관계곱(Fuzzy Relational Product)을 이용한다. 이 퍼지 기법은 백트랙 수를 줄이고 기억장소와 시간의 낭비를 줄여 테스트 패턴 생성의 효율을 증가시킨다.

I. 서론

테스트란 결함이 있는 지점을 찾기위한 과정으로서, 논리회로테스트는 일련의 입력값을 넣어 여기서 출력되는 값이 결함이 없는 회로의 출력값과 같은지 비교함으로써 이루어진다. 여기서 사용된 입력값을 테스트 패턴(test pattern)이라고 한다. n -입력을 가진 조합회로에서 이 회로를 완전히 테스트하려면 2^n 개의 입력을 순서적으로 인가하여 회로를 가동하고 결과를 관찰한다. 따라서 얼마나 적은 테스트 패턴을 사용하여 이 회로를 효율적으로 테스트하는가가 주된 관점이 된다.[1,2] 현재까지 여러 가지 알고리즘과 몇가지 기법들이 제시되어 있다. 그러나 기존의 기법에서는 NP-compelet문제의 팬아웃 지점(fan-out point)에서의 백트랙 문제가 여전히 남아있다. 이러한 문제를 좀더 효과적으로 해결하도록 테스트 생성을 가속화 하기 위해 퍼지 관계논리곱을 이용한다.

퍼지 관계 논리곱(fuzzy relation products)은 조건 연산자(implication operators)와 퍼지 멱집합(fuzzy power sets) 등의 퍼지 특정 기법을 이용하는 것으로 Kohout와 Bandler에 의해 제안된 기법[3,4]이다. 이 기법은 기존의 자료 검색에서는 할 수 없었던 애매한 성질을 갖는 정보의 검색이 가능하다. 이러한 특징을 이용하여 여러 갈래의 선택 상황이 주어졌을 때 가장 효과적인 선택을 하도록 유도할 수 있다.

우선 2장에서는 기존의 테스트 생성 알고리즘을 살펴보고, 3장에서는 퍼지논리곱에 대해, 그리고 퍼지논리곱을 도입한 퍼지 테스트 생성기법을 4장에서, 마지막 결론은 5장에서 보도록 한다.

II. 기존의 테스트 생성 알고리즘.

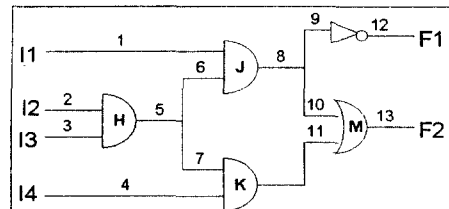
본 논문은 'Stuck-At' 고장 모델[2,5]을 사용한 D-알고리즘[2,5,6]에 기초를 둔다. 어떤 한 지점에서 항상 1

값을 가지는 결함을 Stuck-at-1(s-a-1)이라고 하고 0의 값을 가지는 결함을 Stuck-at-0(s-a-0)라한다.

2-1. D-알고리즘(D-Algorithm)

D-알고리즘은 가장 고전적이고 다른 알고리즘의 기본이 되는 모델이다. D-알고리즘의 기본 개념은 결함이 있는 지점(G0)에서 그다음 게이트(G1, G2, ..., Gn)의 적당한 경로를 선택하는 것으로 먼저 G0에 반대값을 주고 G1, ..., Gn 즉, PO(primary output)까지 따라 나가는 것으로서 Gi에서 입력되는 값을 제외하고는 AND/NAND는 1로, OR/XOR는 0으로 값을 할당시킴으로써 적당한 테스트 패턴을 생성시킬 수 있다.

이러한 과정에서 몇 개의 경로 중 하나를 선택해야 하는 상황이 발생하고, 이때 어느 경로를 선택하느냐에 따라 백트랙수를 최소화하는 관건이 된다. 이러한 선택상황은 @게이트의 입력값으로 여러개의 값중 하나를 선택하는 경우 @팬아웃지점에서 어느 경로에 D(\bar{D})값을 전파할것인가의 두가지 경우로 나눌 수 있다. <그림 1>을 예로 들어보면 다음과 같다.



<그림 1>

위의 그림에서 8번 선에 s-a-0 결함이 발생한다고 가정하면, 이 8번선이 0이 되어야 이 결함을 PO(primary output)로 전파시킬 수 있다. 8번선이 0이 되도록하기 위해서 1번선과 6번선에 00/01/10의 값을 주어야 하는

데, 즉 두 선중 하나만 0이 되면 된다. 이때 어느선에 0을 둘것인가의 선택상황이 발생된다. 또한 8번선이 9번과 10번선으로 나뉘는데 이 지점을 팬아웃지점이라 한다. 이 팬아웃지점을 살펴보면, 8번선에서 발생한 결합이 PO로 전파시키는 과정에서 9번과 10번선중 어느 경로에 \bar{D} 값을 전달할 것인가의 문제가 발생한다. 즉 두라인 중 어느 경로가 백트랙의 확률이 작은가에 따라 테스트 생성 알고리즘의 성능이 좌우된다.

2-2. 기존의 테스트 생성 가속화 기법.

1) 휴리스틱 조건[2]

테스트 용이도(testability measure)를 이용한 방법으로, 테스트 용이도란 회로 내부의 각 신호선들이 외부 입력력선을 통하여 테스트 하기가 어느정도 용이한가를 수치로 제시한 것이다. 한 신호선에 대한 테스트 용이도는 제어도(controllability, CY)와 관측도(observability, OY)로 구성된다. 제어도란 테스트하는 회로의 내부 신호선 값을 PI(primary input)에 의하여 용이하게 제어할 수 있는 정도를 수치로 나타낸 것이고, 관측도는 내부 신호선 값을 PO에 의하여 용이하게 제어할 수 있는 정도를 수치로 나타낸 것이다. 일반적으로 테스트 용이도는 PI로부터 PO순으로 제어도를 우선 계산하고 다시 역으로 PO에서 부터 PI순으로 관측도를 계산하는 두 과정으로 이루어진다. 이러한 테스트 용이도를 설계하는 알고리즘은 SCOAP, CAMELOT 등 여러 가지 기법이 제시되었다.

① SCOAP

제어도와 관측도를 비용 개념으로 생각하여 한 게이트의 출력을 제어하기 위한 비용은 입력을 제어하기 위한 비용으로부터 설계되는 방법으로, CY0과 CY1을 구분하여 사용한다. 입력을 x_1, \dots, x_n , 출력을 z 라 할 때, 각 게이트에 대한 제어도와 관측도의 계산식은 다음과 같다.

초기값 : $CY0(PI) = CY1(PI) = 1$, $OY(PO) = 0$
AND 게이트: $CY0(z) = \text{MIN } CY0(x_i) + 1$.

$$CY1(z) = \sum_{i=1}^n CY1(x_i) + 1$$

$$OY(x_j) = OY(z) + \sum_{i=1, i \neq j}^n CY1(x_i) = 1$$

OR 게이트: $CY0(z) = \sum_{i=1}^n CY0(x_i) + 1$.

$$CY1(z) = \text{MIN } CY1(x_i) + 1$$

$$OY(x_j) = OY(z) + \sum_{i=1, i \neq j}^n CY0(x_i) + 1$$

NAND 게이트: $CY0(z) = \sum_{i=1}^n CY1(x_i) + 1$.

$$CY1(z) = \text{MIN } CY0(x_i) + 1$$

$$OY(x_j) = OY(z) + \sum_{i=1, i \neq j}^n CY1(x_i) + 1$$

NOR 게이트: $CY0(z) = \text{MIN } CY1(x_i) + 1$.

$$CY1(z) = \sum_{i=1}^n CY0(x_i) + 1$$

$$OY(x_j) = OY(z) + \sum_{i=1, i \neq j}^n CY0(x_i) + 1$$

NOT 게이트: $CY0(z) = CY1(x) + 1$.

$$CY1(z) = CY0(x) + 1$$

$$OY(x) = OY(z)$$

② COP

COP는 확률적인 방법을 사용하여 각 신호선의 용이도를 계산하는 기법으로, 즉 제어도 $CY0(CY1)$ 은 신호선이 0(1)값을 갖는 확률을 나타내며, 관측도 OY 는 신호선의 값이 PO에서 관측될 확률을 나타낸다.

각 값은 아래와 같이 계산된다.

$$CY0(PI) = CY1(PI) = 0.5 \quad OY(PO) = 1$$

$$CY1(i) = P(i=1)$$

$$CY0(i) = 1 - CY1(i)$$

$$P_i = \begin{cases} CY0(x_i=0) & 1 \leq i \leq n \\ CY1(x_i=1) & \end{cases}$$

$$CY1(z) = \sum_{i=1}^n \prod_{j=1}^n P_j$$

$$OY(x_i) = OY(z) \cdot \sum_{j=1}^n \prod_{l=1, l \neq j}^n P_l$$

③ CAMELOT

제어도는 PI의 1값에서 시작하여 게이트를 통과할 때마다 CTF(controllability transfer factor)를 사용하여 게이트 출력의 제어도를 새로이 설계한다. 관측도는 PO의 초기치를 1로하여 각 게이트마다 OTF를 사용하여 PI순으로 설계된다. 입력 x_1, \dots, x_n 과 출력 z 를 갖는 게이트에 대한 제어도는 다음과 같다.

$$CTF = 1 - \frac{N(0) - N(1)}{N(0) + N(1)}$$

$$CY(PI) = 1$$

$$CY(z) = CTF \cdot \frac{1}{n} \sum_{i=1}^n CY(x_i)$$

$$OTF = \frac{NP}{NP + NB}$$

$$OY(PO) = 1$$

$$OY(x_i) = OTF \cdot OY(z) \cdot \frac{1}{n} \sum_{i=1}^n CY(x_i)$$

여기서 $N(0)$ 는 게이트가 0을 출력하도록 하는 게이트 입력 조합수이고, $N(1)$ 는 게이트가 1을 출력하도록 하는 게이트 입력 조합수이다.

④ fanout/distance(FD)

회로내 각 신호선의 팬아웃 수만을 고려하여 제어도와 관측도를 설계하는 방법으로 구하는 식은 아래와 같다.

$$IC(i) = \text{팬아웃수}(i) - 1$$

$$CY(i) = IC(i) + \sum_{j=1}^n CY(x_j) \quad (n \text{은 게이트 입력수})$$

$OY = PO$ 로부터의 게이트 레벨 수

FD의 제어도와 관측도는 0이상의 정수값으로 값이 클수록 관측과 제어가 어렵다.

다시 <그림1>을 살펴보자. 각 선에 대한 테스트용이도는 아래 <표1>과 같다.

선택 상황이 발생한 지점을 살펴보자

② 입력값 중 하나를 선택할 경우

1번 선과 6번 선에 대해 보자. 이 두 선에서는 0값에 대한 선택상황이므로 $CY0$ 값을 비교한다. 위의 표에서 보면 SCOAP, FD, CAMELOT에서는 1번 선의 0에 대한 제어가 6번선보다 용이하다는 것을 알 수 있지만, COP에서는 이와 반대의 결과가 나타난다. 그러므로 실제 수행하는 알고리즘에서 COP의 용이도를 사용하면 잘못된 결과값을 얻을 수 있다.

⑤ 팬아웃 지점

8번 선의 \bar{D} 값이 9번과 10번 선중 한 선에 전달되어야 하는데, 이를 위해서는 이 두 선에 대한 OY 값을 비교한다. 위의 표에서 SCOAP, COP, CAMELOT에서는 9번선의 관측이 용이하는 것을 알 수 있으나, FD에서는 이 두선의 관측도가 같다. 이런 경우 기존의

이 두선 중 하나를 임의로 선택하여 알고리즘에서는 사용한다.

신호선	테스트용어도										
	SCOAP			FD			COP			CAMELOT	
	CY0	CY1	OY	CY	OY	CY0	CY1	OY	CY	OY	
1	1	1	5	0	2	0.50	0.50	0.25	1.00	0.50	
2	1	1	5	0	3	0.50	0.50	0.36	1.00	0.27	
3	1	1	5	0	3	0.50	0.50	0.36	1.00	0.27	
4	1	1	7	0	2	0.50	0.50	0.22	1.00	0.06	
5	2	3	3	1	2	0.75	0.25	0.72	0.50	0.55	
6	2	3	3	1	2	0.75	0.25	0.50	0.50	0.50	
7	2	3	5	1	2	0.75	0.25	0.44	0.50	0.10	
8	2	5	1	2	1	0.88	0.13	1.00	0.38	1.00	
9	2	5	1	2	1	0.88	0.13	1.00	0.38	1.00	
10	2	5	3	2	1	0.88	0.13	0.88	0.38	0.19	
11	2	5	3	1	1	0.88	0.13	0.88	0.38	0.19	
12	3	6	0	2	0	0.88	0.13	1.00	0.38	1.00	
13	5	6	0	3	0	0.77	0.23	1.00	0.19	1.00	

<표1>

III. 퍼지 관계곱

논리회로 고장진단을 위한 계층구조(thesaurus)를 생성하기 위해서는 Kohout와 Bandler가 고안한 퍼지 관계논리곱[3,4]을 이용한다. 삼각 논리곱과 사각 논리곱은 아래와 같이 정의한다.

- (1) $(R \triangleleft S)_{ik} = \mu R \triangleleft S(ai,ck) = \pi(aiR \subseteq S_{ck}) = 1/N_j \sum(R_{ij} \rightarrow S_{jk})$
- (2) $(R \triangleright S)_{ik} = \mu R \triangleright S(ai,ck) = \pi(aiR \supseteq S_{ck}) = 1/N_j \sum(R_{ij} \leftarrow S_{jk})$
- (3) $(R \square S)_{ik} = \mu R \square S(ai,ck) = \pi(aiR = S_{ck}) = 1/N_j \sum(R_{ij} \leftrightarrow S_{jk})$
단, $a \leftrightarrow b = \text{MIN}(a \rightarrow b, a \leftarrow b)$

항들 사이의 포함 관계를 나타내는 관계정도 구조를 구성하기 위하여 입력행렬과 그것의 전치행렬의 관계곱을 이용한다. 만약 R이 문항(document)으로부터 항목(term)으로의 관계(D→T)라면 R의 전치행렬 R^T은 항목으로부터 문항로의 관계(T→D)이다. 그러므로 R^T < R은 항으로 부터 항으로의 관계 (T→T)이다.

$$D_{ij} = d_j \text{에 대한 } t_i \text{의 관계 정도} = D_{ij}^T$$

D^T와 D에 대한 삼각논리곱과 사각논리곱의 의미는 아래와 같다.

- $(D^T \triangleleft D)_{ik} = 1/N_j \sum(D_{ij}^T \rightarrow D_{jk})$
: t_i 가 t_k 보다 더 구체적인 정도의 평균등급
- $(D^T \triangleright D)_{ik} = 1/N_j \sum(D_{ij}^T \leftarrow D_{jk})$
: t_i 가 t_k 보다 더 일반적인 평등등급
- $(D^T \square D)_{ik} = 1/N_j \sum(D_{ij}^T \leftrightarrow D_{jk})$
단, $a \leftrightarrow b = \text{MIN}(a \rightarrow b, a \leftarrow b)$
: t_i 와 t_k 가 유사한 정도의 평균등급

위의 퍼지 관계곱을 이용하여 퍼지 테스트 생성 기법을 위한 계층 구조를 형성할 수 있다.

IV. 퍼지 테스트 생성 기법

퍼지 관계논리곱을 이용하여 항목사이의 관계를 고려한 각 항목간의 계층을 구할 수 있게 되는데 이것을 계층구조라고 한다. 삼각 서브 관계 논리곱을 이용할 경우, 이 계층구조를 통하여 어떤 기준에 대한 각 항목간의 우선 순위를 결정할 수 있다. 그러므로 테스트 생성을 위한 새로운 기법에서 항목과 문항를 다음과 같이 재 정의 할 수 있다.

- 문항 = 각 입력선을 체크할 기준
(II절에서 제시한 SCOAP, COP, CAMELOT, FD 값을용용한다.)
- 항목 = 선택상황에 놓여진 각 신호선.
- D_{ik} = d_i 에 대한 t_k 의 속성의 관련정도.

퍼지 입력 행렬은 각 기준을 고려했을 때 백트랙 가능성이 적을수록 높은 퍼지 값을 갖도록 구성한다. 이때 삼각 서브 관계논리곱과 전치행렬[7,8,9,10]을 이용하면 다음의 의미를 갖는 값을 얻게된다.

$(D^T \triangleleft D)_{ik} = t_i$ 가 t_k 보다 좋은 결과에 대한 관련성 정도가 덜함.

즉, 신호선 t_i 가 t_k 보다 백트랙 확률이 높다는 것을 의미하고 현재 제시된 문항를 기준으로 가장 양호한 신호선 순으로 계층을 형성하는 계층구조[10,11]를 얻게 된다.

먼저 각 문항는 II절에서 제시된 휴리스틱의 제어도를 이용하며 0과 1에 대해 표2과 같이 구분 한다.

0값에 대해(D _A)		1값에 대해(D _B)	
d ₁	1/ SCOAP의 CY0	d ₁	1/ SCOAP의 CY1
d ₂	COP의 CY0	d ₂	COP의 CY1
d ₃	CAMELOT의 CY	d ₃	CAMELOT의 CY
d ₄	1/FD의 CY+1	d ₄	1/FD의 CY+1

<표2>

게이트의 입력선이 x₁,x₂,...,x_n이고 출력선이 z인 각 게이트에 대해 살펴보면 다음과 같다.

	z=0	z=1
AND 게이트	행렬구성 term : x ₁ ,x ₂ ,...,x _n document : D _A	x ₁ =x ₂ =...=x _n =1
OR 게이트	x ₁ =x ₂ =...=x _n =0	행렬구성 term : x ₁ ,x ₂ ,...,x _n document : D _B
NAND 게이트	x ₁ =x ₂ =...=x _n =1	행렬구성 term : x ₁ ,x ₂ ,...,x _n document : D _A
NOR 게이트	행렬구성 term : x ₁ ,x ₂ ,...,x _n document : D _A	x ₁ =x ₂ =...=x _n =0
NOT 게이트	x ₁ =x ₂ =...=x _n =0	x ₁ =x ₂ =...=x _n =1

<표3>

팬아웃지점에서는 다음과 같이 문항을 정의한다

팬아웃지점(Dc)	
d ₁	1/ SCOAP의 OY+1
d ₂	COP의 OY
d ₃	CAMELOT의 OY
d ₄	1/FD의 OY+1

<표4>

<그림1>을 예로 들어보자.

③ 1번 선과 6번 선의 경우.

이 선택 문제에 대해 퍼지 기법을 사용하는데, 우선 입력행렬을 구성해야한다. 이 행렬의 문항로 0값에 대한 4가지 값(D_a)과 항목으로 1번과 6번 라인을 둔다.

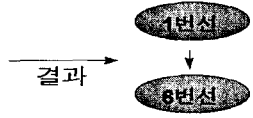
d \ t	1번선	6번선
d ₁	1.00	0.50
d ₂	1.00	0.50
d ₃	0.50	0.75
d ₄	1.00	0.50

<표5>

계산된 결과값과 이 결과값으로 부터 구성되는 계층구조는 그림1과 같다. 이때 퍼지 조건 연산자로 L5 a→b = min(1, 1-a+b) 사용하며, α-cut은 0.9로 한다.

	1번선	6번선
1번선	1	0
6번선	1	1

<표6>



추출된 관계정도 구조에서 1번 라인을 우선적으로 선택한다.

⑥ 9번과 10번선의 경우.

입력행렬의 문항으로 팬아웃지점에 대한 4가지 값(D_c), 그리고 항목으로 9번선과 10번선을 둔다.

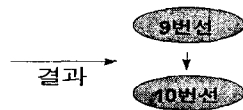
d \ t	9	10
d ₁	0.50	0.25
d ₂	1.00	0.88
d ₃	1.00	0.19
d ₄	0.50	0.50

<표7>

이 결과값으로 부터의 계층구조는 다음과 같다. 이때 사용한 퍼지 조건연산자와, α-cut값은 위와 동일하다.

	9번선	10번선
9번선	1	0
10번선	1	1

<표8>



위의 관계정도 구조로 부터 9번 라인을 우선적으로 선택한다. 위의 결과로 백트랙 수를 줄일 수 있는 가장 적절한 라인을 선택함을 알 수 있다.

V. 결론

퍼지 관계도를 이용한 논리회로의 고장진단을 위한 퍼지 테스트 생성기법은 백트랙 수를 줄일 수 있는 경로를 선택하므로써, 기억공간 및 시간을 절약할 수 있다. 또한 D-알고리즘에서뿐 아니라 그 효율성이 더 높

게 평가되고 있는 또 다른 테스트 생성 알고리즘인 FAN에서 이 기법을 도입한다면 더욱 효과적이라 사료된다. 그러나 입력 행렬의 구성 및 계층구조 형성의 자동화가 앞으로의 과제이다.

참고문헌

- [1] Agrawal, V.D. and Sharad C. seth. Fault Modeling in Test Generation For VLSI Chips pages 15-24, Computer Society Press, 1988
- [2] Brglez F., P.Pownall, and R. Hum, Application of testability analysis : From APTG to critical delay path tracing. Proc. 1984 Int. test Conf., pages 705-712, Oct. 1984
- [3] Bandler, W. and Kohout, L.J. Fuzzy power sets and fuzzy implication operators. In Wang, P.P. and Chang, S.K., editors, *Fuzzy sets and systems*, pages 13-30, plenum press, New York and London, 1980(4).
- [4] Bandler, W. and Kohout, L.J. A survey of fuzzy relational products in their applicability to medicine and clinical psychology. In Bandler, W. and Kohout, L.J., editors, *Knowledge Representation in Medicines and Clinical Behavioural Science*, Abacus Press, Cambridge MA, USA, and Tunbridge Wells, UK, 1986.
- [5] Agrawal, V.D. and Sharad C.seth. Test Generation in Test Generation For VLSI Chips pages 67-77, Computer Society Press, 1988
- [6] Fujiwara, Hideo. Logic testing and design for testability. pages 1-83, The massachusetts Institute of technology, 1985
- [7] Kim, Yong-gi and Kohout, L.j. Use of Fuzzy Relational Products and Algorithms for generaing Control strategies in resolution-based Automated Reasoning. In *IFSA'91 Poceedings*, The International fuzzy Systems Association, 1991
- [8] Kohout, L.J., and Kim, Yong-Gi. generating control strategies for resolution-based theorem provers by means of fuzzy relational products and relational closures. in B. Lowen and M. Roubens, editors, *Fuzzy Logic : State of the Art*. Kluwer Academic, pages 181-192, Boston and Dordrecht, 1993.
- [9] Kim, Yong-Gi and Kohout, L.j. Comparison of Fuzzy Implication Operators by means of Weighting Strategy in Resolution Based Automated Reasoning. In *SAC '92 Proceedings*, pages, ACM symposium on Applied Computing Kansas city, MO, March 1-3,1992.
- [10] Kohout, L.J., Keravnou E. and Bandler, W. Automatics documentary information retrieval by means of fuzzy relational products. In Gaines, B.R., Zadeh L.A. and Zimmermann,H.-J., editors, *Fuzzy Sets in Decision Analysis*, pages 308-404, North-Holland, Amsterdam, 1984
- [11] Kohout, L.J., Keravnou E. and Bandler, W. Information retrieval system using fuzzy relational products for thesaurus construction. In Sanchez, E, and Gupta, M.M, editors, *fuzzy information, Knowledge Reprerentation and Decision Analysis*, pages 7-14, Pergamon Press, Oxford and New York, 1983