컴퓨터공학실험2 5주차 예비보고서

전공: 컴퓨터공학과　　　　　학년: 2　　　　　학번: 20191559　　　　　이름: 강상원

1. De Morgan의 정리에 대해 조사하시오.

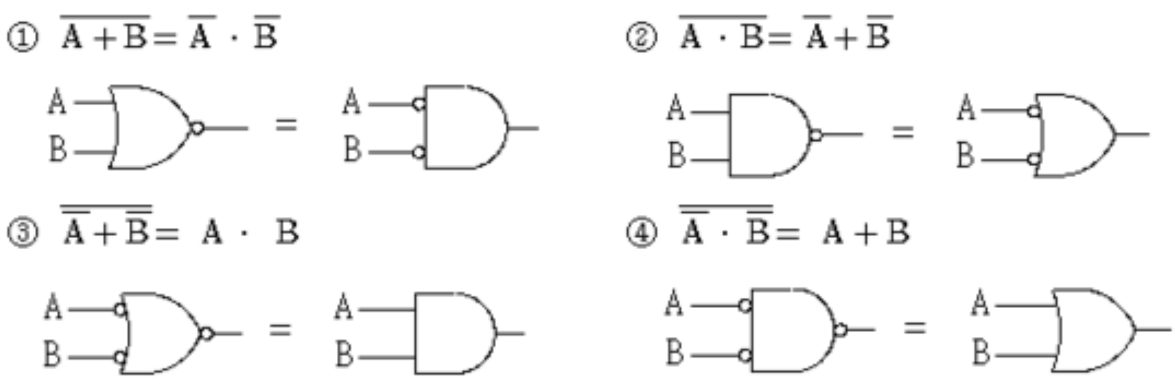
드모르간의 정리는 크게 아래의 두 식으로 정리될 수 있다.

$$\overline{x \cdot y} = \bar{x} + \bar{y}$$
$$\overline{x + y} = \bar{x} \cdot \bar{y}$$

집합 표기 방식으로 나타낸다면 다음과 같다.

$$\overline{(A \cap B)} = \overline{A} \cup \overline{B}$$
$$\overline{(A \cup B)} = \overline{A} \cap \overline{B}$$

첫번째 식은 '변수들의 곱의 부정은 각각의 변수의 부정을 취해 합한 것과 같다', 두번째 식은 '변수들의 합의 결과에 보수를 취한 것은 각각의 변수를 부정하여 곱한 것과 같다'로 풀이할 수 있다.
이를 논리회로의 등가성으로 표현한다면 다음과 같이 나타낼 수 있다.



드모르간의 정리의 증명은 진리표를 이용해 할 수 있다.

1) $\overline{x \cdot y} = \bar{x} + \bar{y}$

x	y	$x \cdot y$	$\overline{x \cdot y}$	$\bar{x}$	$\bar{y}$	$\bar{x} + \bar{y}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

2) $\overline{x+y} = \bar{x} \cdot \bar{y}$

x	y	$x+y$	$\overline{x+y}$	$\bar{x}$	$\bar{y}$	$\bar{x} \cdot \bar{y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

녹색으로 표시한 부분을 비교하면 알 수 있듯이 1)과 2) 식 모두 성립한다.

2. 논리회로의 간소화에 대해 조사하시오(예시 포함).

논리회로의 간소화는 복잡해 보이는 논리회로를 비교적 더 간단한 회로로 만들어준다. 이를 적용하는 대표적 두 가지 방법, 카르노 맵과 콰인-맥클러스키 알고리즘에 대해 3, 4번에서 후술하겠다. 또 기본적으로 부울 대수를 이용하는 방법도 있다.

입력값 전부와 출력값이 동일한 경우에는 논리 회로를 가능한 간단하게 만드는 것이 유리하다.

∵ 논리 회로의 소비 전력, 동작 속도 향상

논리회로의 간소화를 적용한 예시는 다음과 같다.

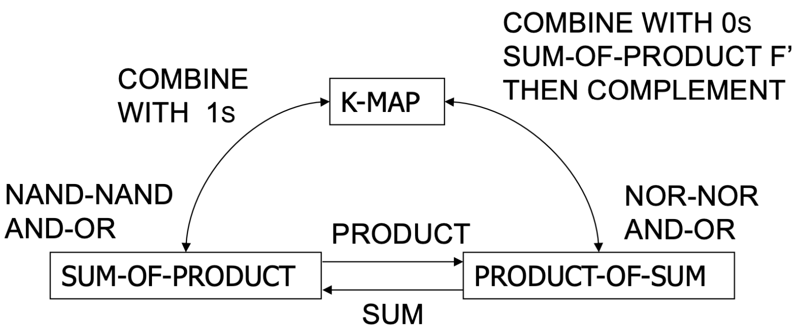
(A’+B+C’+D) (A+B’+C’+D’)(A+B’+C’+D) (A+B’+C+D’) (A’+B’+C’+D)(A+B+C’+D)

= (A+B’+D’)(C’+D)

3. 카르노 맵에 대해 조사하시오(예시 포함).

카르노 맵은 시각적으로 논리식을 표현해 그 회로식을 간소화하는 방법이다. 가장 작은 식을 도출해 낸다는 보장은 없다. + 입력값이 6개가 넘어가면 카르노 맵으로 최소화하기 어렵다.

카르노 맵을 이용해 간략화된 논리식에는 Essential Prime Implicant 전부, non-Essential Prime Implicant 중 일부가 포함된다.



↑카르노 맵의 변환 구조 도식화

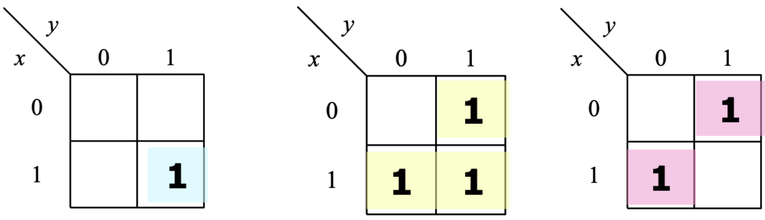
카르노 맵은 함수에서 각각의 최소항에 대하여 한 개의 사각형으로 구성되어 있다.
∴ 2변수 카르노 맵은 4개의 사각형, 3변수 카르노 맵은 8개의 사각형.. 을 가진다.

어떠한 논리식(함수)를 카르노 맵으로 그릴 때, 대응하는 각 사각형에 1을 기입하고 포함되지 않는 사각형은 비워 두거나 0을 쓴다. don't care의 경우에는, X를 기입한다. 아래는 2변수 카르노 맵의 예이다.

x	y	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

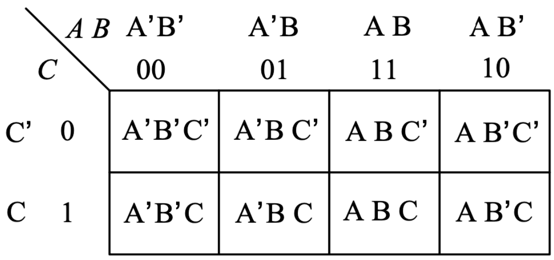
(a) Truth table

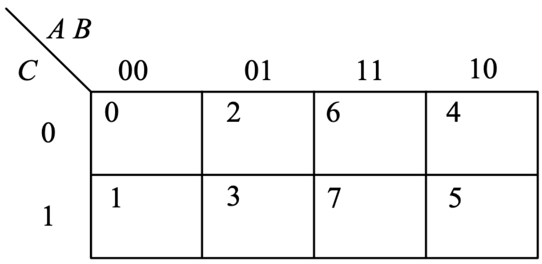
이러한 진리표로 표현되는 논리식(함수)를



(b) AND: xy (c) OR: $x + y$ (d) XOR: $x'y + xy'$

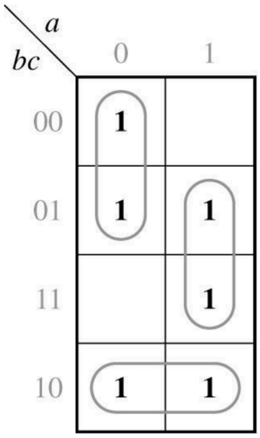
위와 같이 표현 가능하다.
3변수 카르노 맵부터 주의해야 할 사항이 생기는데, 아래의 맵에서 보이듯이



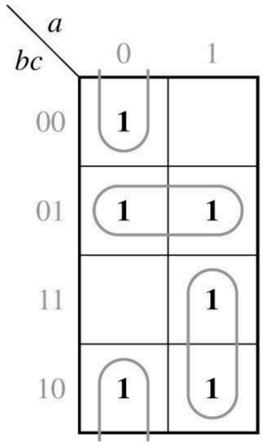


마지막 2개의 열이 일반적인 순서와 다름을 알 수 있다. 인접한 사각형은 원소가 하나씩만 다르도록 (공통점이 있도록) 배열한다. 이렇게 하면 차후 인접한 사각형을 결합할 때 수월해진다.
+ 공통인 부분끼리 묶을 때는 2^n개씩 묶어야 한다. 최대한 많은 수의 공통끼리 묶고, 모서리 부분은 반대편으로 연결되는 성질도 활용한다. 이웃해 있는 항끼리 묶어야 하며, 반드시 직사각형, 정사각형 형태로 묶어야 한다. 적용 예시는 다음과 같다.

$F = \sum m(0,1,2,5,6,7)$일 때,



$F = a'b' + bc' + ac$



$F = a'c' + b'c + ab$

위와 같이 표현 가능하다.

이와 같은 방식을 sum of product(SOP)라 하며 다른 방식인 product of sum(POS)는 반대로 묶으면 된다.

4. Quine-McCluskey 최소화 알고리즘에 대해 조사하시오.

콰인-맥클러스키 최소화 알고리즘은 변수가 6개 이상일 때 유용하다. SOP 방식으로 만들어지며, 입력된 진리표에서 최소항을 찾아 그 중 입력항의 1 개수에 따라 그룹을 만든다. 1비트씩 다른 항들응 찾아 간소화하고, 더 이상 간소화되지 않을 때까지 이를 반복한다.

이 과정이 끝나면 Prime Implicant(PI) 항들이 나오게 되고, 이 PI 차트를 활용해 최소 개수의 SOP 논리식을 찾는다.

예시)

$f(a,b,c,d) = \sum m(0,1,2,5,6,7,8,9,10,14)$일 때,

다음과 같이 2진법으로 변환한 다음 그룹화한다.

group 0	0	0000
group 1	1	0001
	2	0010
	8	1000
group 2	5	0101
	6	0110
	9	1001
	10	1010
group 3	7	0111
	14	1110

	Column I	Column II	Column III
group 0	0 0000 ✓	0, 1 000– ✓	0, 1, 8, 9 –00–
group 1	1 0001 ✓	0, 2 00–0 ✓	0, 2, 8, 10 –0–0
	2 0010 ✓	0, 8 –000 ✓	~~0, 8, 1, 9 –00–~~
	8 1000 ✓	1, 5 0–01	~~0, 8, 2, 10 –0–0~~
group 2	5 0101 ✓	1, 9 –001 ✓	2, 6, 10, 14 -- 10
	6 0110 ✓	2, 6 0–10 ✓	~~2, 10, 6, 14 -- 10~~
	9 1001 ✓	2, 10 –010 ✓	
	10 1010 ✓	8, 9 100– ✓	
group 3	7 0111 ✓	8, 10 10–0 ✓	
	14 1110 ✓	5, 7 01–1	
		6, 7 011–	
		6, 14 –110 ✓	
		10, 14 1–10 ✓	

줄일 수 있는 출력을 don’t care로 바꾸면서 줄인다. 이러한 방법으로 PI를 구할 수 있다.

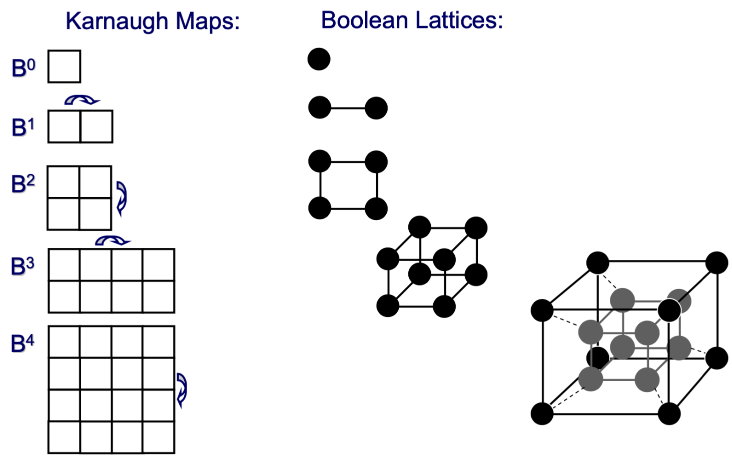
EPI를 구하는 방법은 다음과 같은 방식으로 진행된다.

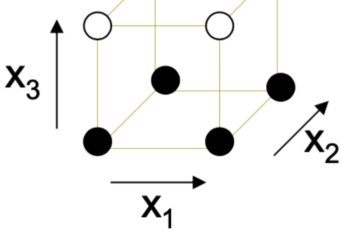
여기서 원 식은 $f(A,B,C,D) = \sum(0,1,2,3,5,7,8,10,12,13,15)$이다.

	0	1	2	3	5	7	8	10	12	13	15
$\bar{A}\bar{B}$	X	X	X	X							
$\bar{B}\bar{D}$	X		X				X	X			
$\bar{A}D$		X		X	X	X					
BD					X	X				X	X
$A\bar{C}\bar{D}$							X		X		
$AB\bar{C}$									X	X	
Essential	X		X		X	X	X	X		X	X

5. 기타이론.

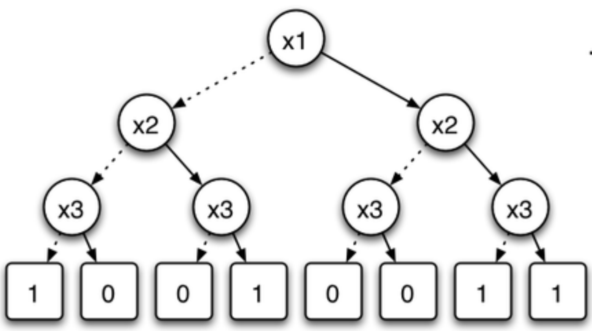
Boolean Space를 카르노 맵 이외에도 큐브 형태로 나타낼 수 있다.



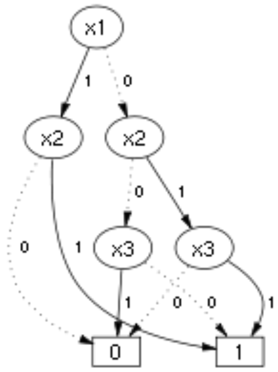


$x_1x_2x_3$		f
0 0 0		1
0 0 1		0
0 1 0		1
0 1 1		0
1 0 0	⇒	1
1 0 1		0
1 1 0		1
1 1 1		0

또는 Binary Decision Diagram(BDD)으로 논리식을 나타낼 수 있는데, 예시는 다음과 같다.



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



여러 의사결정 노드와 터미널 구조로 구성된 그래프로 나타낼 수 있다. 터미널 노드에는 0, 1 두 종류가 있다. 뻗어나가는 가지는 점선, 실선으로 표현된다. 위 BDD는 $f(x_1, x_2, x_3) = \overline{x_1}\,\overline{x_2}\,\overline{x_3} + x_1x_2 + x_2x_3$를 나타낸 것이다.