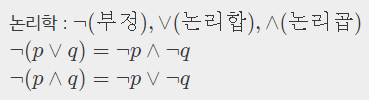
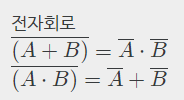
1. 드 모르간의 법칙은 집합론이나 수리논리학 그리고 컴퓨터 사이언스 등에서 논리곱, 논리합, 부정 연산들의 관계를 정리한 것이다.

<논리학>

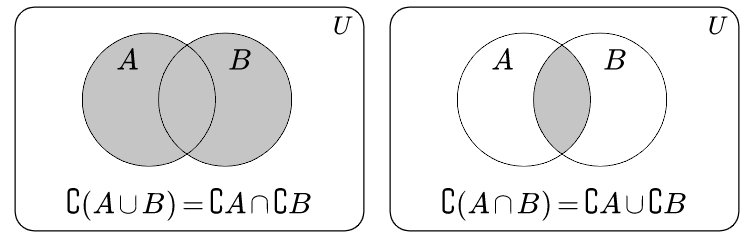


<전자회로>



이는 표기법은 서로 조금씩 다르다. 하지만 내용은 동일하다.

벤다이어그램으로 보면 둘이 같다는 것을 잘 알 수 있다.



A U B의 여집합은 A의 여집합과 B의 여집합의 교집합과 같은 것을 알 수 있고, 오른쪽 벤다이어그램을 보면, A와 B의 교집합 전체에 여집합한 것과 A여집합과 B여집합의 합집합은 같은 것을 알 수 있다.

컴퓨터 과학에서 사용하는 Boolean algebra에 대해 드 모르간의 법칙을 적용할 수 있다. (A+B)’ = A’B’가 되고, (AB)’ = A +B’가 된다. 즉, 드 모르간의 법칙을 적용하면 SOP로 표현된 식을 POS로 표현한 식보다 더 쉽게 변경할 수 있다는 장점이 있다.

2. 논리회로를 간소화하는 방법으로는 Boolean algebra를 이용하거나 카르노 맵을 이용할 수 있다. 논리회로를 간소화하면 논리회로의 개수를 줄이면서 속도를 향상시킬 수 있다는 장점이 있고, 비용적으로도 감소시킬 수 있다는 장점이 있다

**<switching algebra>**

(commutative)

P1a. a+b = b+a

P1b. a∙b = ba

(Associative)

P2a. a + (b+c) = (a + b) +c

P2b. a(b∙c) = (a∙b)c

(identity)

P3a. A + 0 = a

P3b. a∙1 = a

P3aa. 0 + a = a

P3bb. 1∙a = a

(NULL)

P4a. a + 1 = 1

P4b. a∙0=0

P4aa. 1 + a = 1

P4bb. 0∙a = 0

(Complement)

P5a. a + a’ = 1

P5b. a∙a’ = 0

P5aa. a’ + a = 1

P5bb. a’∙a = 0

(Idempotency)

P6a. a + a = a

P6b. a∙a = a

(involution)

P7. (a’)’ = a

(Distributive)

P8a. a(b + c) = ab + ac

P8b. a + bc = (a + b)(a + c)

(Adjacency)

P9a. ab + ab’ = a

P9b. (a+b)(a + b’) = a

(Simplfication)

P10a. a + a’b = a + b

P!0b. a(a’ + b) = ab

(DeMorgan)

P11a. (a + b)’ = a’b’

P11b. (ab)’ = a’ + b’

P11aa. (a+b+c…)’ = a’b’c’…

P11bb. (abc…)’ = a’ + b’ + c’

(Absorption)

P12a. a + ab = a

P12b. a(a+b) = a

(Consensus)

P13a. at1 + a’t2 + t1t2 = at1 + a’t2

P13b. (a + t1)(a’ + t2)(t1 + t2) = (a + t1)(a’ + t2)

P14a. ab + a’c = (a + c)(a’ + b)

예시를 보면 다음과 같이 간소화할 수 있다.

예시) (A+B)’C+AB’

= A’B’C + AB’ ->Demorgan

= B’A’C + B’A ->Commutative

= B’A + B’A’C ->Commutative

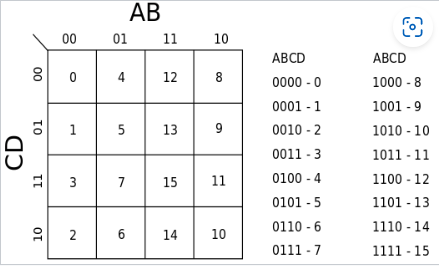
= B’(A+A’C) -> Distributive

= B’(A+A’)(A+C) -> Distributive

= B’(A+C) -> Complement

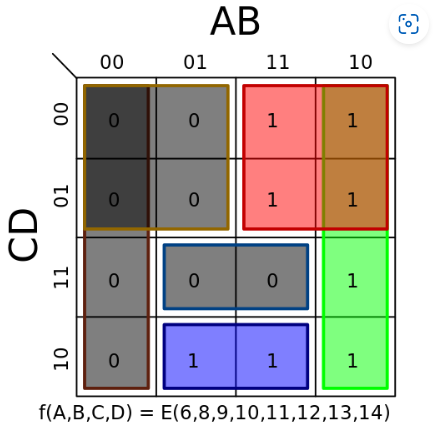
= B’(A+C) -> Identity

3. 카르노 맵은 불 대수 위의 함수를 단순화할 때 사용하는 맵이다. 불대수에서 확장된 logic 표현들을 직접 체크하면서 연관된 상호관계를 이용해서 간소화하는 방법이다.



카르노 맵의 구조는 위와 같다.

예시로 진리표의 출력 값을 임의로 배치하면, 다음과 같이 배치할 수 있다.



서로 이웃한 출력끼리 묶어서 상자를 만든다. 같은 출력의 패턴을 찾아서 묶음을 만들 수 있다. 따라서 단순화해보면, 빨간 사각형의 경우 변수 A의 경우 1일 때 1로 그대로 출력되고, C의 경우 0으로 같은 출력이 되므로, AC’로 출력된다. 녹색상자는 AB’이고, 파란상자는 BCD’이다. 즉, 원래의 식은 ABC’D’ + ABC’D + AB’C’D’ + AB’C’D + AB’CD + ABCD’ + A’BCD’ + ABCD’ 이었지만, 카르노맵을 사용하면 AC’ + AB’ + BCD’으로 간소화할 수 있다.

4. Quine-McCluskey 알고리즘은 논리식을 최소화를 시킬 수 있는 알고리즘이다. 내부만 보면 카르노 맵과 동일하지만 맵을 쓰는 카르노 맵과는 달리 표를 사용하여 컴퓨터에서도 쉽게 쓸 수 있다. 그리고 논리함수의 최소 형태를 결정론적으로 구한다. 과정은 먼저 주어진 함수의 후보항들을 모두 구하고, 후보항들을 이용해서 후보항 표에서 EPI를 구해주면 된다. 카르노 맵과는 달리 Quine-McCluskey 알고리즘은 변수가 4개가 넘어도 효율적으로 논리를 최소화 및 최적화를 할 수 있다는 장점이 있다.

5. Minterm은 모든 변수가 항상 한 번씩 사용된 product term이다. 또한 maxterm은 모든 변수가 한 번씩 항상 사용된 sum term이다. 그리고 Sum of product는 SOP라고 쓰며, disjunctive normal form이고, POS는 product of sum으로 conjunctive normal form이다. 그리고 minimum SOP는 최소한의 product terms를 이용해서 SOP로 표현한 것이다.