컴퓨터공학실험2 5주차 예비 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**0. 목차**

1. De Morgan의 정리를 조사한다.

2. 논리 회로의 간소화에 대해 조사한다.

3. 카르노 맵에 대해서 조사한다.

4. Quine-McCluskey 최소화 알고리즘에 대해 조사한다.

**1. De Morgan의 정리**

De Morgan의 정리는 19세기 수학자인 오거스터스 드 모르간이 발견해낸 논리학과 집합론에서 중요한 정리이다. 논리합과 논리곱에 대해서 여집합의 적용을 정리한 것이다.1 회로 이론의 입장에서 본다면, NAND gate와 Negative-OR gate가 동일하며, NOR gate와 Negative-AND gate가 동일하다는 정리이다. 그 형태는 아래와 같다.

(1) (X∩Y)c = Xc∪Yc (1) ~(XY) = (~X)+(~Y)

(2) (X∪Y)c = Xc∩Yc (2) ~(X+Y) = (~X)(~Y)

<집합론에서 드모르간 법칙> <회로 이론에서 드모르간 법칙>

**2. 논리 회로의 간소화**

논리 회로를 실제로 만들 때는, 논리 회로의 비용이 사용되는 gate의 개수와 직결하게 된다. 즉, gate의 개수를 줄이는 것이 회로를 제작할 때 중요한 요소가 된다. 예를 들자면, 위에서 말한 De Morgan의 법칙에서 (X+Y)’ = X’ Y’ 라 할 수 있는데, 왼쪽 표현은 OR, NOT 2 개의 gate로 구현할 수 있다면, 오른쪽 표현은 2개의 NOT gate와 1개의 AND gate가 필요하기 때문에 왼쪽 표현으로 회로를 만들어내는 것이 더 효율적이라고 할 수 있다. 여기에서 오른쪽 표현을 왼쪽 표현과 같이 gate수를 적게 만들어내는 것이 논리 회로의 간소화 라고 할 수 있다.

논리 회로의 간소화를 할 때에, 대표적으로 아래와 같은 정리들을 토대로 간소화를 할 수 있다.

1. X+X = X (자기 자신의 OR gate는 자기 자신이다.)

2. XX = X (자기 자신의 AND gate는 자기 자신이다.)

3. XX’ = 0 (자신과 complement의 AND gate는 0이다.)

4. X + X’ = 1 (자신과 complement의 OR gate는 1이다.)

5. 1X = 1 (1과 자신의 AND gate는 자신이다.)

6. 0X = 0 (0과 자신의 AND gate는 0이다.)

7. 1 + X = 1 (1과 자신의 OR gate는 1이다.)

8. 0 + X = X (0과 자신의 OR gate는 자신이다.)

9. X(Y+Z) = XY + XZ (분배 법칙)

10. XY + XZ = X(Y + Z) (결합 법칙)

등이 존재한다. 추가적으로 위에서 설명한 드모르간 법칙 등도 간소화를 할 때 사용할 수 있다.

이를 실제로 예시를 들어 적용해보자.

[AB’(C + BD) + A’B’]C 라는 식을 간소화해보자. 우선 분배법칙을 통해 C를 분배하면

AB’C(C + BC) + A’B’C, 좌항을 다시 분배법칙을 이용하면

AB’CC + ABB’CC + A’B’C 이다. 정리 2를 통해 AB’CC = AB’C, 3을 통해 ABB’CC = 0 따라서

AB’C + A’B’C 이다. 이를 B’C로 결합하면

(A + A’)B’C, 정리 4에 의해서 A + A’ = 1 따라서

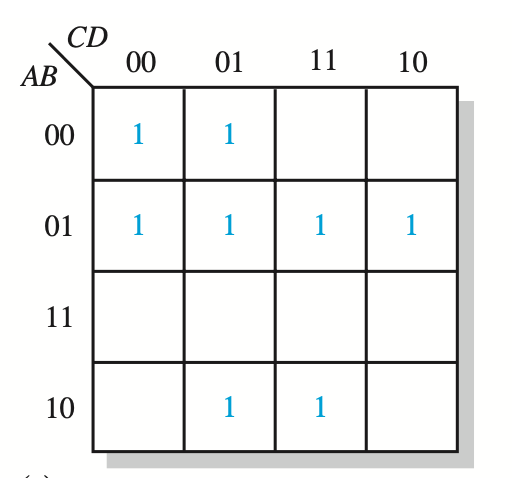
B’C 로 간소화할 수 있다.

위 과정을 통해서 처음에는 9개의 gate를 필요로 하는 논리식이 2개의 gate만 이용하면 되는 식으로 표현됨을 파악할 수 있다.

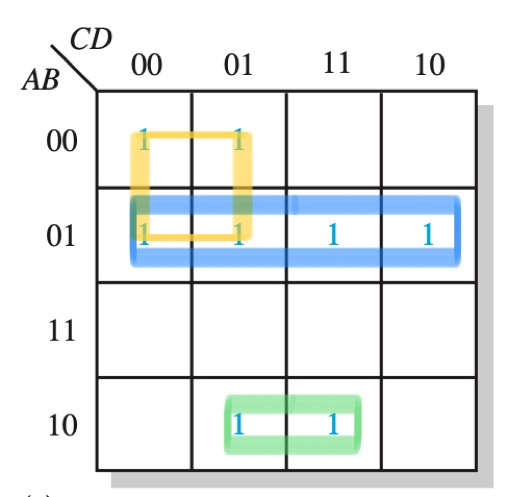
**3. 카르노 맵**

위에서 설명한 것처럼, 간소화는 회로를 제작할 때 중요한 요소이다. 하지만 복잡한 논리 회로를 간소화해내는 것은 쉽지만은 않다. 추가적으로 간소화를 통해서 요약된 논리 회로를 얻어냈다고 해도, 구해낸 회로가 gate가 최소인 회로인지 알아기 힘들기도 하다. 이를 쉽게 구해낼 수 있는 하나의 방법이 카르노 맵(Karnaugh map)이다. 카르노 맵이란 변수를 행렬 형태로 표현한 이후에, 결과가 1인 것들에 대해서 가장 큰 사각형들의 집합으로 나타내는 것이 카르노 맵 방법이다. 카르노 맵 방법을 사용하면 결과적으로는 SOP(Sum of Product) 형태로 여러 AND gate들의 결과를 마지막으로 OR gate에 적용하는 형태를 얻어낼 수 있다. 카르노 맵을 적용시키는 예시는 아래와 같다.

아래와 같은 map2을 만들었다고 하자.(아래 map은 input A, B, C, D의 진리표라고 생각하면 된다. 사각형들을 쉽게 만들기 위해 AB, CD 에 대한 4x4 진리표를 만들어낸 것이다.)



이를 여러 사각형들의 분할로 만들어내면 된다. 이는 아래와 같다.



이후에는 ABCD와 그 complement들에 대해서 값이 1이 되는 형태를 만들어내면 된다.

우선 노란색 사각형에서는 A가 0, C가 0인 부분이므로, A’C’ 이다. 파란색 부분은 A’B 이며, 초록색 부분은 AB’D 이다.(0이든 1이든 상관없는 형태를 지워준다.) 이를 결과적으로 정리하면 A’C’ + A’B + AB’C 임을 알아낼 수 있다. 이렇게 카르노 맵을 통해서 간소화를 진행할 수 있다.

**4. Quine-McClusky 최소화 알고리즘**

Quine-McClusky 최소화 알고리즘 또한 카르노 맵과 비슷하게 논리 회로를 간소화(최소화)할 수 있는 하나의 방법이다. Quine-McClusky 최소화 알고리즘의 가장 큰 장점은 표를 그린 이후 사각형 그림들을 통해서 논리식을 최소화하는 카르노 맵과 다르게, 컴퓨터를 통해서 이 최소화된 논리식을 확정적으로 구할 수 있다는 점이다.

이 알고리즘을 수행하기 위해서 우선 (1) minterm(최소항)을 파악해야 한다. 여기에서 최소항은 변수와 변수의 역에 대해 곱연산(AND gate)의 값이 1이 되는 상태를 의미한다. 예를 들어서, ABC 3 input에 대해서 A’BC = 1, ABC = 0, A’B’C = 1 이라는 결과가 나왔다면, minterm을 m1->011, m2->001 이라고 할 수 있다. 여기에서 minterm들을 SOP 형태로 나타낸다면, fA,B,C = A’BC + A’B’C 임 또한 알아낼 수 있다. 이 이후에는 (2) 최소항들의 결합을 수행해야 한다. 여기에서 기본적으로 사용되는 법칙은 위의 예시에서 A’BC와 A’B’C에서 B, B’에서 차이가 존재하는데, B가 1, 0임에 관계없이 값은 A’C에서 결정됨을 알아낼 수 있다. 이를 통해서 minterm을 계속해서 결합하며, (3) 결합이 끝난 이후 남겨진 minterm에 대해서 새로운 표를 만들어내며(Prime Implicants table), 이때 모든 minterm을 포함할 수 있는 prime implicants(최소항들의 결합을 통해 만들어진 식)들의 합이 논리 회로를 최소화 하는 식이 된다.

**6. 출처**

1. Thomas L. Floyd, Person, Digital Fundamentals Eleventh edition(global edition), 199p

2. Thomas L. Floyd, Person, Digital Fundamentals Eleventh edition(global edition), 227p