5주차 예비보고서

전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

**1.**

De morgan의 정리는 AND 연산자를 OR 연산자로, OR 연산자를 AND 연산자로 바꾸며, 피연산자는 NOT 연산을 한다. 식으로 표현하자면 ~(a & b) = (~a | ~b) 혹은 ~(a | b) = (~a & ~b)와 같이 표현할 수 있다. 이때 ~(a | b) = (~a & ~b)를 드 모르간의 제 1법칙, ~(a & b) = (~a | ~b)을 드 모르간의 제 2법칙이라 한다. 드 모르간의 1법칙은 두 변수에 대한 NOR 연산이 두 변수 각각의 NOT 연산을 AND 연산한 결과와 같다. 드 모르간 2법칙은 두 변수에 대한 NAND 연산이 두 변수 각각의 NOT 연산을 OR 연산한 결과와 같다. 드 모르간의 법칙을 진리표로 정리하면 두 식이 같다는 것을 알 수 있다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | ~a | ~b | a&b | a|b | ~a&~b | ~a|~b | ~(a&b) | ~(a|b) |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

위 진리표에서 파란색 하이라이트 부분은 드 모르간의 제 1법칙이, 초록색 하이라이트 부분은 드 모르간의 제 2법칙이 참이 됨을 보이고 있다.

**2.**

논리회로의 간소화는 여러개의 논리 회로를 적은 수의 논리회로로 간단히 하는 것을 의미한다. 논리회로를 간소함으로써 회로 구현에 필요한 논리 회로의 개수를 줄일 수 있어 효율적인 회로 구성을 가능하게 하며, 복잡한 구성을 간단히 할 수 있어 회로를 파악하기 용이해진다. 이러한 논리회로 간소화 방법에는 대수적으로 간소화 하는 방법(불 대수 법칙)과 알고리즘을 통해 간소화하는 방법(카르노 맵, quine-McCluskey 최소화 알고리즘) 등이 있다.

불 대수 법칙은 논리식과 같은 결과를 가지는 다른 논리식으로 변경 가능하다는 점에서 논리회로를 간소화할 수 있다. 이를 통해서 논리항의 개수를 감소시키거나 문자의 수를 감소시킨다. 불 대수 법칙은 다음과 같다.

1. a + b = b + a
2. a +(b + c) = (a + b) + c
3. a + 0 = a
4. a + 1 = 1
5. a + ~a = 1
6. a + a =a
7. ~(~a) = a
8. a ⋅ b = b ⋅ a
9. a ⋅ (b ⋅ c) = (a ⋅ b) ⋅ c
10. a⋅ 1 = a
11. a ⋅ 0 = 0
12. a ⋅ ~a = 0
13. a ⋅ a = a
14. a ⋅ b + a⋅ c = a ⋅ (b +c)
15. a ⋅ b + a ⋅ (~b) = a
16. ~a ⋅ ~b + ~a ⋅ b + a ⋅ b + a ⋅ ~b = 1
17. a + ~a ⋅ b = a + b
18. ~a ⋅ ~b = ~(a + b)
19. a + a ⋅ b = a
20. a ⋅ b + ~a ⋅ c = (a + c) ⋅ (~a +b)
21. (~a + ~b) ⋅ (~a + b) ⋅ (a + b) ⋅ (a + ~b) =0
22. a ⋅ (~a + b) = a ⋅ b
23. ~(a ⋅ b) = ~a + ~b
24. a ⋅ (a + b) =a

이 중 몇몇 법칙은 논리회로의 간소화를 구현할 수 있다. 예를 들어 a ⋅ 1 =a 라는 식은 두개의 논리항을 하나의 논리항으로 간소화할 수 있다. 뿐만 아니라 진리표에 의해 두 논리항은 모든 경우에 참임을 보이고 있다.

|  |  |  |
| --- | --- | --- |
| a | 1 | a ⋅ 1 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

~a ⋅ ~b + ~a ⋅ b + a ⋅ b + a ⋅ ~b = 1 법칙도 7개의 논리 연산을 간소화 할 수 있으며, 식 간소화를 이용하여 두 논리항이 참임을 나타낼 수 있다.

~a ⋅ ~b + ~a ⋅ b + a ⋅ b + a ⋅ ~b = ~a ⋅ ( ~b +b) +a ⋅ (b + ~b) …… 14번 법칙에 의해

= ~a ⋅ 1 + a ⋅ 1 …… 5번 법칙에 의해

=~a + a …… 10번 법칙에 의해

= 1 ……5번 법칙에 의해

다음과 같은 식도 간소화가 가능하다.

a⋅b⋅c + (~a) ⋅b + (~a) ⋅(~b)

=a⋅b⋅c + (~a)

=b⋅c + (~a)

**3.**

불 대수 법칙과 같은 대수적 간소화 방법은 간소화 한 논리식이 최소값인지에 대해 보장할 수 없다. 그에 대한 대안으로 알고리즘 형식으로 논리회로를 간소화하고자 하는 방법이 카르노 맵을 이용하는 것이다. 카르노 맵은 Sum of Product에서 적절한 곱의 항을 찾는 도식적인 방법론이다. 카르노맵은 입력값과 출력값을 진리표로 표현한 뒤, 카르노 맵을 표현한다. 카르노 맵은 변수들을 표로 정리하여야 하는데 다음과 같은 규칙을 따라야 한다. 변수 x에 대해 x가 참이면 1, 거짓이면(NOT x가 참) 0으로 표현하여야 한다. 또한 비트를 표현할 떄, 1비트 씩 차이가 나도록 한다. 00-01-10-11 순서가 아닌 00-01-11-10으로 표현한다. 이렇게 된다면 00-01, 01-11, 11-10, 10-00이 모두 하나의 비트만 바꾸는 것으로 표현이 가능하다.

2변수 카르노맵을 예제를 통해 구현하였다.

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

변수 A,B로 이루어진 논리식 A’B’+A’b를 간소화하고자 한다. A’B’는 두 변수가 모두 0일때이므로, A=0, B=0에 1을 적는다. A’B도 같은 방식으로 A=0, B=1에 1을 적는다. 이 두 1을 묶는다면, a는 0일때, b값에 관계없이 참이되므로, 논리식은 A’가 된다.

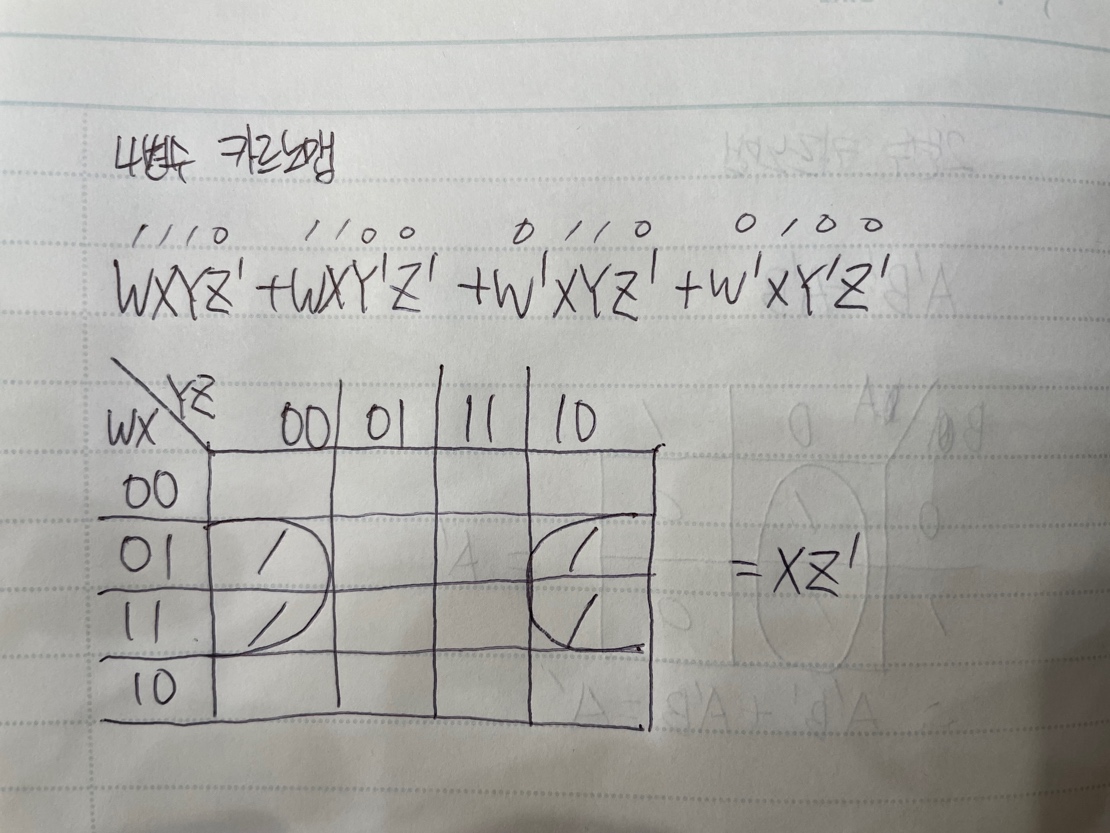
3변수 카르노맵을 예제를 통해 구현하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

3개의 변수 A,B,C로 이루어진 논리식을 간소화하고자 한다. A와 B를 묶어서 표를 만든다. 이후 식에 해당되는 칸에 1을 적는다. ABC’(A=1,B=1,C=0), AB’C’(A=1,B=0,C=0), A’BC’(A=0,B=1,C=0), A’B’C’(A=0,B=0,C=0)가 해당한다. 1인 칸을 묶으면 C=0일때, A,B의 값과 관계없이 참을 얻을 수 있다. 따라서 해당 논리식을 간소화한 식은 C’이다.

4변수 카르노맵을 예제를 통해 구현하였다.



4개의 변수 W,X,Y,Z로 이루어진 논리식을 간소화하고자 한다. W,X를 묶고, Y,Z를 묶어 표를 만든다. 이후 식에 해당하는 칸에 1을 적는다. WXYZ’(1110), WXY’Z’(1100), W’XYZ’(0110), W’XY’Z’(0010)이 해당된다. 이후 1이 적힌 칸끼리 묶는데, 앞서 설명했듯이 1비트씩 변화가 되도록 표를 구성하여 00과 10이 묶일 수 있다. 이때 X는 1일때만, Z는 0일때만, W와 Y값에 관계없이 참이 되므로, 해당 논리식을 간소화한 값은 XZ’이 된다.

**4.**

Quine-McCluskey 최소화 알고리즘은 논리회로를 간소화하는 알고리즘으로 컴퓨터가 이용 가능하다는 특징이 있다. 카르노 맵이 그림을 이용하기 때문에, 컴퓨터가 이용하기 어렵지만 Quine-McCluskey 최소화 알고리즘은 표를 이용하여 컴퓨터도 이용 가능하다. Quine-McCluskey 최소화 알고리즘은 다음과 같은 순서로 진행된다. 우선 주어진 논리식의 prime implicants를 구한다. Implicant는 논리식에 대해 sum of product를 표현하는 데에 사용할 수 있는 곱의 항을 의미하며, Prime implicants는 다른 implicant와는 결합되지 않는 implicant를 의미한다. 이후 prime implicants를 활용하여 표에서 essential prime implicants를 구한다. Essential prime implicant는 다른 prime implicant에 포함되지 않는 1을 포함하는 prime implicant를 의미한다. 이후 남은 prime implicants를 계산한다면 최소화된 논리식을 얻을 수 있다.

**5.**

논리식을 sum of product(곱의 합) 혹은 product of sum(합의 곱)으로 표현한 정규형은 Minterm(최소항), Maxterm(최대항)의 전개로 표현할 수 있다. 우선 minterm의 전개는 논리식의 모든 변수를 한 번씩 사용된 곱의 합 형태로 표현한 것이다. 구하는 방법은 논리식이 1이 되는 모든 경우의 변수들의 곱의 합을 구하여 얻는다. Maxterm의 전개는 논리식의 모든 변수를 한 번씩 사용한 합의 항을 표현한 것이다. 구하는 방법은 논리식이 1이 되는 모든 경우의 변수들의 합을 곱하여 얻는다.

**6. 참고문헌**

Alan B. Marcvitz, Introduction to Logic Design, McGraw-Hill(2010)

T. K. Jain, D. S. Kushwaha and A. K. Misra, "Optimization of the Quine-McCluskey Method for the Minimization of the Boolean Expressions," Fourth International Conference on Autonomic and Autonomous Systems (ICAS'08), 2008

Wang, C., Tao, Y. Karnaugh Maps of Logical Systems and Applications in Digital Circuit Design. *Circuits Syst Signal Process* **39**, 2245–2271 (2020).