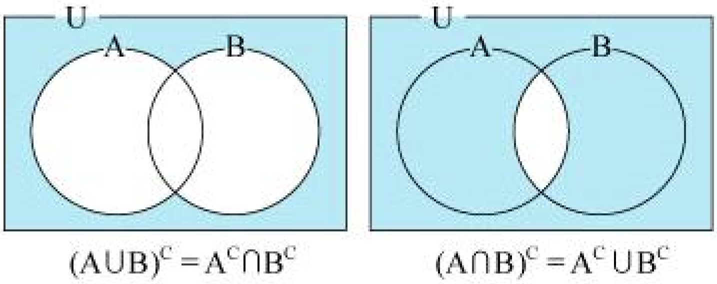
5주차 예비보고서

전공: 신문방송학과　　학년: 4학년　　학번: 20191150　　이름: 전현길

1. De Morgan의 정리에 대해 조사하시오.



드 모르간의 법칙은 **논리합의 부정은 부정의 논리곱과 같으며, 논리곱의 부정은 부정의 논리합과 같다**는 법칙이다. 벤 다이어그램, 집합 기호로는 위처럼 표현할 수 있고, 부울 대수(boolean algebra)로는 다음과 같이 표현한다.

$$(A + B)' = A'B',\ (AB)' = A' + B'$$

디지털 회로에서는 더 이상 간소화할 수 없는 최소 합의곱(minimum POS), 최소 곱의합(minimum SOP) 식을 찾은 뒤 서로 바꿀 때 드 모르간의 법칙을 이용할 수 있다.

2. 논리회로의 간소화에 대해 조사하시오.

논리회로를 서술하는 방식은 여러 가지가 있다. **1) 일상 언어로 서술**할 수도 있고, **2) 진리표를 사용**할 수도 있으며, **3) 대수식을 사용**할 수도 있다.

진리표만으로도 우리가 원하는 동작을 하는 회로를 구현하는 데에는 충분하지만(ROM), 실제 필요한 회로보다 훨씬 더 복잡한 회로를 구현하게 될 수 있다. 회로가 복잡해질수록 회로 면적, 지연, 전력 효율성이 나빠지므로 회로를 가능한 한 간소화시키는 것이 디지털 회로 공학의 목표이다.

이를 위해서 **회로를 입력에 의한 출력 함수 형태로 표현하여, 가능하다면 함수식을 간소화할 수 있도록 한 방법**이 부울 대수(boolean algebra; switching algebra) 체계이다. 스위칭 대수는 3가지 논리 연산(AND, OR, NOT)을 시작으로 스위칭 대수의 정리(theorem)들을 유도하며 정의된다. 회로 설계자는

스위칭 대수의 다양한 정리들을 바탕으로 논리회로를 간소화시킬 수 있다. 아래에 자주 쓰이는 theorem들을 정리해 보았다.

Switching Algebra		
Commutative	a + b = b + a	ab = ba
Associative	a + (b + c) = (a + b) + c	a(bc) = (ab)c
Distributive	a(b + c) = ab + ac	a + bc = (a + b)(a + c)
Identity	0 + a = a + 0 = a	a · 1 = 1 · a = a
Null	a + 1 = 1 + a = 1	a · 0 = 0 · a = 0
Complement	a + a’ = a’ + a = 1	a · a’ = a’ · a = 0
Idempotency	a + a = a	a · a = a
Involution	(a’)’ = a	
Absorption	a + ab = a	a(a + b) = a
De Morgan’s	(a + b)’ = a’b’	(ab)’ = a’ + b’
Adjacency	ab + ab’ = a	(a + b)(a + b’) = a
Simplification	a + a’b = a + b	a(a’ + b) = ab
Consensus	ab + bc + ca’ = ab + ca’	(a+b)(b+c)(c+a’) = (a+b)(c+a’)

3. 카르노 맵에 대해 조사하시오.

위의 부울 대수식을 활용하여 임의의 함수를 간소화할 수 있지만, 몇 가지 문제가 있다. 형식화된 공식이 존재하지 않으므로 **1) 경험론적인 방법에 의지해야 한다**는 점, 함수를 더 이상 조작하지 못하는 상태에 이르렀더라도 **2) 함수가 최소화되었다고 확신할 수 없다**는 점이다.

카르노 맵(Karnaugh Map)은 **부울 대수식을 좀 더 쉽고, 형식적으로 간소화시키기 위해 이진 함수를 시각적으로 표현하는 방법**이다. 카르노 맵은 진리표의 각 항들을 하나의 정사각형으로 대응시키는데, 따라서 2변수 맵은 4개의 정사각형, 3변수 맵은 8개, 4변수 맵은 16개의 정사각형을 갖게 된다. 각 **정사각형에는 입력값에 따른 함수식의 출력값을 쓰고, 회로에서 출력이 어찌 되든 상관없는 부분(don’t care항)은 X를 쓴다**. 카르노 맵은 3개, 4개의 변수를 가진 대부분의 문제를 효율적으로 해결할 수 있고, 6개의 변수를 가진 문제까지 유용하게 쓰일 수 있다.

AB / CD	00	01	11	10
00	0	0	1	1
01	1	1	1	X
11	1	1	1	X
10	0	0	0	0

카르노 맵으로 대수식을 간소화시키는 방법은 단순하다. minterm인 정사각형 2^n개를 직사각형으로 가능한 한 크게 묶어서, 직사각형의 개수가 최대한 적도록 하는 것이 원칙이다.

예를 들어, 위의 4변수 카르노 맵의 경우 f = D + AC'로 축약할 수 있는데, 이는 don't care항을 무시하고 가장 큰 직사각형을 그렸을 때 AC', D라는 두 직사각형이 나타나기 때문이다.

4변수 카르노 맵을 통해 대수식을 간소화시킬 때 가장 주의해야 할 부분은 B'D' 항이다. B'D' 항은 카르노 맵의 각 모서리를 모두 포함하는 항으로, 카르노 맵에서 놓치기 쉽지만 대수식을 간소화시킬 때 중요할 수 있는 항이다.

4. Quine-McCluskey 최소화 알고리즘에 대해 조사하시오.

Quine-McCluskey 최소화 알고리즘은 함수의 주내포항(prime implicant)을 찾기 위한 방법으로, 최소항을 먼저 구한 뒤 인접 속성(adjacency property)을 반복적으로 사용해 주내포항을 구하는 방식이다. 표를 이용한 간소화가 가능하기 때문에, 카르노 맵과 달리 5-6개 이상의 변수를 갖더라도 효과적이고 컴퓨터에서 쉽게 실행할 수 있다. 또한 최소화된 논리식을 결정론적으로 구할 수 있다.

알고리즘은 다음과 같은 과정을 거쳐 논리식을 최소화한다. 그림은 위키백과를 참고하였다.

$$f(A, B, C, D) = \Sigma(4, 8, 9, 10, 11, 12, 14, 15)$$

1) SOP로 표현: 진리표를 작성하여 최소항(minterm)들을 찾고, 곱의 합(canoncial sum of products) 형태로 표현한다.

	section	후보항	bit 표기	pi
크기가 1인 후보항	1	m4	0100	
		m8	1000	
	2	m9	1001	
		m10	1010	
		m12	1100	
	3	m11	1011	
		m14	1110	
	4	m15	1111	
크기가 2인 후보항	1	m(4,12)	-100	*
		m(8,9)	100-	
		m(8,10)	10-0	
		m(8,12)	1-00	
	2	m(9,11)	10-1	
		m(10,11)	101-	
		m(10,14)	1-10	
		m(12,14)	11-0	
	3	m(11,15)	1-11	
		m(14,15)	111-	
크기가 4인 후보항	1	m(8,9,10,11)	10--	*
		m(8,10,12,14)	1--0	*
	2	m(10,11,14,15)	1-1-	*

2) 후보항을 탐색: 각 최소항들을 최소항 표에 넣는다. 이 때 처음 넣는 항들은 크기가 1인 후보항이다. 각 최소항이 한 자리만 차이난다면, 그 자리를 대체해 크기가 이전보다 2배 큰 후보항으로 표현한다. 더 이상 후보항을 합치지 못할 때까지 후보항을 합친다. 이 때, **더 이상 후보항을 합치지 못하는 후보항은 주항(prime implicant)**이다.

	4	8	9	10	11	12	14	15
m(4,12)*	X					X		
m(8,9,10,11)*		X	X	X	X			
m(8,10,12,14)*		X		X		X	X	
m(10,11,14,15)*				X	X		X	X

3) 후보항 표로 표현: 주항을 찾았으므로 최소화된 식을 얻기 위해 후보항 표를 작성한다. 세로로는 찾은 주항들을 나열하고, 가로로는 앞에서 얻었던 최소항을 나열한다. 모든 X를 표현할 수 있는 경우 최소화된 식이다. 이 경우 1, 2, 4 또는 1, 3, 4를 선택하는 경우 최소화된 식을 얻을 수 있다.

후보항 표에서 필수 주항(essential prime implicant) 역시 쉽게 찾을 수 있는데, 세로로 중복되어 나타나지 않는 X를 갖는 주항이 필수 주항이다. 예의 경우 m(4, 12)와 m(10, 11, 14, 15)가 그러한 X를 가지므로 필수 주항이다.

5. 기타 이론

대수식 간소화에 관련하여 아래와 같은 용어들을 추가적으로 정리해 보았다.

용어 정리

- **이진함수(boolean function):** B = {0, 1}, variable: x_1, x_2
- **letter:** 상수 또는 변수, ex) x_1, x_2, 0, 1
- **literal:** letter과 letter의 보수, ex) x_1, x_2, 0, 1, (x_1)', (x_2)'
- **terms(항)**
  - **Product term:** 1 또는 **서로 다른 letter들로 구성된** 상수가 아닌 리터럴의 교집합
    - **'서로 다른 letter들로 구성된'의 의미** → 1, x, xy, y'z 등은 성립, xyx', xyy'는 성립 x
    - **Minterm:** 모든 변수가 항상 한 번씩 사용된 product term(곱항)
  - **Sum term:** 0 또는 서로 다른 letter들로 구성된 상수가 아닌 리터럴의 합집합
    - **Maxterm:** 모든 변수가 항상 한 번씩 사용된 sum term(합항)
- **Sum of Product(SOP, DNF; disjunctive normal form)**
  - Canonical SOP: 어떤 이진함수의 모든 minterm의 합
- **Product of Sum(POS, CNF; conjunctive normal form)**
  - Canonical POS: 어떤 이진함수의 모든 maxterm의 곱
- **Implicant(묶음):** 카르노 맵에서 2^k개의 1의 묶음
  - **Prime Implicant:** 더 큰 implicant에 포함되지 않는 implicant
  - **Essential Prime Implicant**
    - 이진함수를 표현하는데 꼭 필요한 prime implicant
    - **단 하나의 PI에만 포함되는 민텀을 갖고 있는** prime implicant
  - 간략화된 이진함수는 essential prime implicant를 전부 포함하고 있음