5주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

De Morgan의 정리는 두 가지 식으로 나타낼 수 있다. 두 식 모두 부정을 취할 때 AND와 OR이 서로 바뀐다는 특징이 있다. 첫 번째 식은 ~(A\*B) = (~A) + (~B)이다. 이는 제 2법칙으로 A\*B의 보수가 A의 보수와 B의 보수의 합과 같다는 뜻이다. 두 번째 식은 ~(A+B) = (~A)\*(~B)이다. 이는 1법칙으로 A+B의 보수는 A의 보수와 B의 보수의 곱과 같다는 뜻이다. 각각의 집합기호를 이용해서도 나타낼 수 있는데, AND는 교집합 기호로, OR는 합집합 기호로 나타낼 수 있다. 또한 부정의 경우는 여집합으로 표기할 수 있다. 드 모르간의 법칙을 증명하는 방법으로는 우선 집합론의 방식으로 증명하는 방식이 있다. 2법칙으로 증명을 해보면 x가에 속하는 원소이면, x는 A∩B에 속하지 않는 것이므로 A나 B에 모두 속하지 않는다. 이는 A의 여집합, B의 여집합 중 하나에 속한다는 의미이고, 이는 라 할 수 있으므로 성립하는 것을 확인할 수 있다. 1법칙 역시 비슷한 방식으로 증명을 할 수 있다. 이외에는 진리표의 확인을 통해서 증명을 할 수 있는 방법도 있다.

................

**2.**

.......................

논리회로의 간소화는 복잡하게 작성되어 있는 논리회로를 같은 결과를 보여주는 더 간단한 회로로 만들어주는 것을 의미한다. 3, 4번에 나오는 방식을 따르면 좀 더 빠르게 간소화를 진행할 수 있다. 그러나 논리식에서 적용되는 법칙들을 사용하여 직접 정리하는 방식을 사용할 수도 있다. 이 때 논리식을 나타내는 방식을 불대수라고 한다.

예시를 이용하면 F=A’B’C’ + A’B’C + A’BC’ + AB’C + ABC’ + ABC가 주어졌다고 하면 각각 우선 결합 법칙과 교환 법칙에 의해서 F=A’B’(C’ + C) + (A’ + A)BC’ + A(B’ + B)C로 우선 줄일 수 있고, 이를 보수 법칙을 통해 정리하면 F=A’B’ + BC’ + AC가 된다.

.........................

**3.**

.......................

카르노 맵은 논리식을 간소화하는 방법 중 하나이다. 이는 변수의 개수에 따라 그리는 방법이 달라지는데, 각 변수마다 0 또는 1의 값을 가지고 있고, 나올 수 있는 경우의 수를 2차원 형태로 나타내므로 2변수의 경우 4칸, 3변수의 경우 8칸, 4변수의 경우 16칸과 같은 방식으로 경우를 나눠준다. 사용하는 방식으로는 변수들이 가질 수 있는 경우의 수로 표를 만들어준 후, 각각의 항에 대해서 부정인 경우 0으로, 원래대로인 경우 1을 채워넣는다. 그 후 표에서 인접해있는 1을 묶었을 때 변하지 않는 변수를 확인하여 묶어주면 된다. 카르노 맵을 통해서 식을 정리했을 때 나온 식이 항상 가장 간소화된 식인지는 확실하지 않으므로 정리 후에 확인해야 할 필요성도 존재한다. 또한 변수가 6개 이상일 경우에 대해서는 카르노 맵으로 간소화하기 어렵다는 점도 있다.

2변수인 경우와 3변수인 경우에 대해 예시를 통해서 알아본다. F = a’b’ + a’b를 먼저 간소화해본다. 이는 2변수이므로 4칸의 표가 필요하다.

|  |  |  |
| --- | --- | --- |
| a\b | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

이는 a’b’이므로 a=0, b=0인 칸에 1을, a’b이므로 a=0, b=1인 칸에 1을 작성하고 나머지 칸은 0으로 채운다. 이를 확인하면 인접한 1 두개를 묶을 수 있는데, 여기서 a’는 고정되어 있고 b와 b’으로만 바뀌는 것을 확인할 수 있고, 여기서 고정된 값인 a’만 남기면 간소화된 식이 된다. 다음으로는 3변수에 대해서 알아본다. 위에서 사용한 F=A’B’C’ + A’B’C + A’BC’ + AB’C + ABC’ + ABC를 이용하기로 했다. 표를 그리면 아래와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A\BC | 00 | 01 | 11 | 10 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |

위와 같이 그리면 우선 ABC가 000인 경우와 001인 경우를 보면 A’B’이 고정되어 있다. 따라서 여기는 A’B’만 남고, 다음으로 101과 111을 보면 AC가 고정되어 있으므로 AC, 마지막으로 110과 010의 경우에는 BC’이 고정되어 있으므로 BC’이 남아서 최종 정리하면 F = A’B’ + AC + BC’이 된다.

.........................

**4.**

.......................

Quine-McCluskey 최소화 알고리즘은 6변수 이상에서 논리식을 간소화할 때 사용하는 알고리즘이다. 이 알고리즘은 크게 두 단계로 나뉘는데, 우선 주어진 식에서 후보항들을 모두 구하는 과정이 있고, 그 다음 이들을 이용해서 필수항을 구하는 과정이 있다. 각 항들을 카르노 맵에서 변수들에 대해 이진법으로 원래 형태와 부정형을 구분한 것처럼 이진법으로 변환한다. 그 후 항들을 1의 개수에 따라서 그룹으로 묶어준다. 이들이 최소항인데, 이들 중 한 자리씩 차이나는 경우에는 해당 겹치는 자리를 -로 대체하며 결합한다. 더 이상 어떤 항도 결합할 수 없을 때 필수항 표를 만든다. 표에 후보항들이 어떤 최소항들을 나타낼 수 있는 지 정리한 후, 모든 최소항을 나타낼 수 있도록 선택하면 간소화를 마칠 수 있다.

.........................

**5.**

.......................

NAND와 NOR를 이용하여 논리 게이트를 나타낼 수 있다. 한 종류로 모든 논리 표현을 나타낼 수 있으므로 이를 Universal gate라고 한다. 이를 확인하는 방법은 하나의 게이트로 NOT, AND, OR을 모두 만들 수 있는 경우에 모든 Boolean expression을 만들 수 있으므로 확인해보면 된다. 우선 NAND로 NOT은 두 입력 모두 X로 넣으면 ~X가 출력되는 것을 확인할 수 있다. AND는 두 입력 X, Y로 NAND 연산을 수행한 출력 값을 다음 NAND에 두 입력에 모두 전달하면 구현된다. OR은 드 모르간의 법칙을 이용하여 두 입력 X, Y에 각각 NAND로 구현한 NOT 처리를 해준 후에 그 출력 2개를 NAND에 전달하면 된다. 다음으로 NOR로 NOT은 NAND와 똑같이 두 입력에 입력 X를 넣어주면 ~X가 출력된다. 다음으로 AND는 NAND로 OR을 구현할 때와 같은 방식으로 구현하면 된다. OR은 NAND로 AND를 구현하는 방식과 같은 방식으로 구현하면 된다.

.........................