5주차 예비보고서

전공 : 컴퓨터공학과 학년 : 3학년 학번 : 20211558 이름 : 윤준서

**1. De Morgan의 정리**

De Morgan의 정리는 어떤 논리식을 전체 Interval을 취한 형태는, AND는 OR, OR은 AND로 바꾸고 각각의 변수에 Interval을 취한 형태와 같음을 나타낸다. De Morgan의 정리에는 두 가지 법칙이 있는데, 다음과 같다.

제 1법칙 : --> **~(A + B) = ~A • ~B**

제 2법칙 :  --> **~(A • B) = ~A + ~B**

De Morgan의 정리는 두 개의 변수 간의 연산과 더불어 여러 개의 변수에도 똑같이 적용할 수 있다. 변수가 4개인 경우에도 제 1법칙 **~(A + B + C + D) = ~A • ~B • ~C • ~D** 가 성립한다. 이는 제 2법칙에서도 똑같이 성립한다.

De Morgan의 정리는 Interval 상태에서도 성립하는데, **~(~A + ~B) = A • B** 로 유사하게 나타낼 수 있다.

**2. 논리회로의 간소화**

논리회로의 간소화란 기존의 논리식을 결과 값은 같지만 더 간단한 형태로 나타내는 것을 뜻한다. 물리적인 공간에 제약이 있는 논리회로는 사용에 필요한 트랜지스터, 게이트의 개수가 적을 수록 속도와 성능에 유리하므로 비용을 줄이되 같은 결과를 얻을 수 있는 논리식을 사용하는 것이 중요하다.

논리식을 간소화할 수 있는 여러 기본 법칙과 공식이 존재하는데, 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **Commutative** | **A + B = B + A** | **A • B = B • A** |
| **Associative** | **A + (B + C) = (A + B) + C** | **A • (B • C) = (A • B) • C** |
| **Identity** | **A + 0 = A** | **A • 1 = A** |
| **Null** | **A + 1 = 1** | **A • 0 = 0** |
| **Complement** | **A + A' = 1** | **A • A' = 0** |
| **Idempotency** | **A + A = A** | **A • A = A** |
| **Involution** | **(A')' = A** | |
| **Distributive** | **A • (B + C) = A • B + A • C** | **A + B • C = (A + B) • (A + C)** |
| **Adjacency** | **A • B + A • B' = A** | **(A + B) • (A + B') = A** |
| **Simplification** | **A + A' • B = A + B** | **A • (A' + B) = A • B** |
| **De Morgan** | **(A + B)' = A' • B'** | **(A • B)' = A' + B'** |
| **Absorption** | **A + A • B = A** | **A • (A + B) = A** |
| **Consensus** | **A • B + A' • C = (A + C) • (A'+ B)** | |

위의 공식을 통해 De Morgan의 정리 또한 논리회로의 간소화에 사용되는 것을 알 수 있다.

논리회로의 간소화의 예시로 **Y = A + A•B' + A•C•D + A•E'•F•G+A•H•I --> Y = A** 가 있다.

Absorption 법칙을 사용하여 기존의 식을 아주 간단하게 줄일 수 있다.

**3. 카르노 맵**

카르노 맵(Karnaugh Map)은 논리식을 간소화하는 하나의 방법이다. 카르노 맵은 표를 그려 해당 식이 최소로 간소화되었는지를 직관적으로 확인할 수 있다는 장점이 있다. 논리회로의 변수는 0과 1중 하나만을 값으로 가지므로 카르노 맵의 칸 개수는 2의 제곱수가 된다.

식 **Y = A' • B' + A • B** 을 예시로 2변수 카르노 맵을 그리면 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **B**  **A** | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

변수가 A, B 두 개이므로 칸의 개수는 2의 제곱인 4칸인 것을 알 수 있다. 카르노 맵은 변수들의 값에 따라 각 식의 결과 값을 맵에 작성한 형태다.

카르노 맵을 이용해 식을 간소화하기 위해서 우선 표의 값이 1인 부분들을 묶는다. 즉 초록색으로 표시하면 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **B**  **A** | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

초록색으로 표시된 두 칸 만을 보면 A'는 값이 일정하고 B와 B'가 나타나는 것을 알 수 있다. 즉 A'가 고정된 값이므로, 기존의 논리식을 **Y = A'** 로 가장 짧게 간소화할 수 있다.

맵에서 결과 값이 1인 경우들 중, 고정된 값의 변수만을 사용해 논리식을 최소한으로 표현하는 것이 카르노 맵의 목적이다. 위의 경우처럼 1을 기준으로 묶으면 합의 간소화가 되고, 0을 기준으로 묶으면 곱의 간소화가 된다.

**4. Quine-McCluskey 최소화 알고리즘**

위에서 알아본 카르노 맵은 변수가 5개 이상이 되면 그만큼 표가 복잡해 지며 최소로 간소화된 식을 찾기가 어려워지는 단점이 있다. 변수의 개수가 5개 이상일 때 식을 간소화하기 위한 방법으로 Quine-McCluskey 최소화 알고리즘을 사용할 수 있다. Quine-McCluskey 알고리즘은 카르노 맵과 내부적으로는 동일하나 표를 사용해 코드로 작성할 수 있다는 점과, 최소로 간소화된 식을 보장하는 장점이 있다.Quine-McCluskey 알고리즘의 과정은 **PI 식별 단계**와 **PI 선택 단계**로 나눌 수 있다.

**논리식 Y = A'C'D + A'CD' + B'CD' + AB'D' + A'BD + BC'D + A'BC + AB'C** 을 Quine-McCluskey 알고리즘을 통해 간소화 시키는 과정은 다음과 같다.

**<PI 식별 단계>**

1. 우선 논리식을 Boolean Function으로 표현한다.

2. 논리식의 변수들의 값에서 1의 개수에 따라 인덱스를 부여하고 그룹을 생성한다.

|  |  |  |
| --- | --- | --- |
| **1의 개수** | **최소항** | **ABCD** |
| **1** | **m1** | **0001** |
| **m2** | **0010** |
| **m8** | **1000** |
| **2** | **m5** | **0101** |
| **m6** | **0110** |
| **m10** | **1010** |
| **3** | **m7** | **0111** |
| **m11** | **1011** |
| **m13** | **1101** |

3. 각 그룹의 항을 비교해 한 비트만 다른 항들을 찾아 간소화한다.

|  |  |  |
| --- | --- | --- |
| **1의 개수** | **크기가 2인 항** | **ABCD** |
| **1** | **m(1, 5)** | **0-01** |
| **m(2, 6)** | **0-10** |
| **m(2, 10)** | **-010** |
| **m(8, 10)** | **10-0** |
| **2** | **m(5, 7)** | **01-1** |
| **m(5, 13)** | **-101** |
| **m(6, 7)** | **011-** |
| **m(10, 11)** | **101-** |

간소화가 끝난 항들을 PI(주항, Prime Implicants)라고 부른다.

4. 중복되는 PI를 찾기 위해 표를 새로 만든다. 이때 중복되지 않는 PI를 EPI(필수주항, Essential Prime Implicants)라고 부른다.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PI** | **ABCD** | **m1** | **m2** | **m5** | **m6** | **m7** | **m8** | **m10** | **m11** | **m13** |
| **A'BC** | **011-** |  |  |  | **O** | **O** |  |  |  |  |
| **AB'C** | **101-** |  |  |  |  |  |  | **O** | **O** |  |
| **AB'D'** | **10-0** |  |  |  |  |  | **O** | **O** |  |  |
| **A'BD** | **01-1** |  |  | **O** |  | **O** |  |  |  |  |
| **A'C'D** | **0-01** | **O** |  | **O** |  |  |  |  |  |  |
| **A'CD'** | **0-10** |  | **O** |  | **O** |  |  |  |  |  |
| **B'CD'** | **-010** |  | **O** |  |  |  |  | **O** |  |  |
| **BC'D** | **-101** |  |  | **O** |  |  |  |  |  | **O** |

**노란색**으로 칠한 PI가 중복되지 않는 PI를 포함하는 **EPI**를 가진다.

5. EPI에 포함되는 PI를 없애고 EPI가 포함하지 않는 최소항을 구한다. 위의 표를 참고하면 EPI가 포함하지 않는 최소항은 **m2, m6, m7**임을 알 수 있다. 이들을 사용해 간소화된 논리식을 얻을 수 있다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PI** | **ABCD** | **m2** | **m6** | **m7** |
| **A'BC** | **011-** |  | **O** | **O** |
| **A'BD** | **01-1** |  |  | **O** |
| **A'CD'** | **0-10** | **O** | **O** |  |
| **B'CD'** | **-010** | **O** |  |  |

위의 PI를 위에서부터 차례대로 1, 2, 3, 4로 번호를 붙이면, m2, m6, m7을 모두 최소로 포함하는 (1, 3) (1, 4) (2, 3)를 통해 세 가지 최소 논리식을 얻을 수 있다.

**(1, 3) -> Y = AB'C + AB'D' + AC'D' + BC'D + A'BC + A'CD'**

**(1, 4) -> Y = AB'C + AB'D' + AC'D' + BC'D + A'BC + B'CD'**

**(2, 4) -> Y = AB'C + AB'D' + AC'D' + BC'D + A'BD + A'CD'**

파란색 항은 EPI이고, 검은색 항은 없애고 EPI가 포함하지 않는 최소항이다.

**5. 기타이론**

다음은 카르노 맵과 퀸-맥클러스키 이외에 논리식을 간소화하는 알고리즘이다.

**• Espresso 알고리즘** : 다수의 입력 변수를 가진 복잡한 논리식을 효율적으로 간소화한다. 퀸-맥클러스키 알고리즘보다 계산 속도가 빨라 대규모 논리 회로의 최적화에 적합하다.

**• BDD(이진 결정 다이어그램)** : 논리식을 그래프 형태로 표현한다. 논리식의 구조를 효율적으로 표현하고 간소화하는 데 사용되며, 대규모 논리 회로에서 유용하다.