

mc-cluskey algorithm 결과 보고서.

row dominance와 column dominance 구현했음.

```
{---start solve-----
-----start-----
pi: ['0-1--', '0--0-', '0---0', '110-1', '11-1-', '-01--', '-0-01', '-1-10', '--001', '--11-']
minterms: 0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 17, 20, 21, 22, 23, 25, 26, 27, 30, 31,
-----
findEPI:
      0   1   2   4   5   6   7   8   9   10  12  13  14  15  17  20  21  22  23  25  26  27  30  31
0-1--  -   -   -   0   0   0   0   -   -   -   0   0   0   0   -   -   -   -   -   -   -   -   -
0--0-  0   0   -   0   0   -   -   0   0   -   0   0   -   -   -   -   -   -   -   -   -   -   -
0---0  0   -   0   0   -   0   -   0   0   -   0   -   0   -   -   -   -   -   -   -   -   -
110-1  -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   0   -   0   -   -
11-1-  -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   0   0   0   0
-01-   -   -   -   0   0   0   0   -   -   -   -   -   -   -   0   0   0   0   -   -   -   -
-0-01  -   0   -   -   0   -   -   -   -   -   -   -   -   0   -   0   -   -   -   -   -   -
-1-10  -   -   -   -   -   -   -   -   -   0   -   -   0   -   -   -   -   -   -   0   -   0   -
--001  -   0   -   -   -   -   -   0   -   -   -   -   -   0   -   -   -   -   0   -   -   -
--11-  -   -   -   -   -   0   0   -   -   -   -   0   0   -   -   -   0   0   -   -   -   0   0

epi: ['0---0', '-01--']
after del epi minterms: 1, 9, 13, 15, 17, 25, 26, 27, 30, 31,
nepi:['0-1--', '0--0-', '110-1', '11-1-', '-0-01', '-1-10', '--001', '--11-']
```

1. epi 찾기. 특정 minterm을 유일하게 감싸고 있는 pi를 찾는다. epi와 nepi를 출력한다.

```
column dominance start:
pi: ['0-1--', '0--0-', '110-1', '11-1-', '-0-01', '-1-10', '--001', '--11-']
minterms: 1, 9, 13, 15, 17, 25, 26, 27, 30, 31,
      0-1--  0--0-  110-1  11-1-  -0-01  -1-10  --001  --11-
1   -   0   -   -   0   -   0   -
9   -   0   -   -   -   -   0   -
13  0   0   -   -   -   -   -
15  0   -   -   -   -   -   0
17  -   -   -   0   -   0   -
25  -   -   0   -   -   -   0   -
26  -   -   -   0   -   0   -
27  -   -   0   0   -   -   -
30  -   -   -   0   -   0   -   0
31  -   -   -   0   -   -   -   0

1>=9
30>=26
after column dominance minterms: 9, 13, 15, 17, 25, 26, 27, 31,
```

2. column dominance

column dominance를 찾는다. 표의 좌측에 minterm들이 놓여 있고 상단에 pi가 놓여 있기 때문에 눈으로 볼 때는 가로로 감싸는 형태를 찾으면 된다.

1>=9는 1이 9를 포함한다는 뜻이다. 부분집합 기호를 비슷하게 표현하였다.

두 집합 중 하나가 다른 하나에 포함되는 형태일 때, 포함하는 쪽 집합을 제거한다. 즉 1>=9이면 pi들을 담고 있는 집합 1을 제거하는 것이다.

맨 아랫줄에 보면 1과 30이 사라졌다는 것을 알 수 있다.

```

row dominance:
      9  13  15  17  25  26  27  31
0-1-- -  0  0  -  -  -  -  -
0--0-  0  0  -  -  -  -  -
110-1 -  -  -  -  0  -  0  -
11-1- -  -  -  -  -  0  0  0
-0-01 -  -  -  0  -  -  -  -
-1-10 -  -  -  -  -  0  -  -
--001  0  -  -  0  0  -  -  -
--11- -  -  0  -  -  -  -  0

11-1->=-1-10
--001>=-0-01
after row dominance nepi: ['0-1--', '0--0-', '110-1', '11-1-', '--001', '--11-']
-----start-----
pi: ['0-1--', '0--0-', '110-1', '11-1-', '--001', '--11-']
minterms: 9, 13, 15, 17, 25, 26, 27, 31,
-----

```

3. row dominance

column dominance와 마찬가지로 눈으로 볼 땐 가로로 감싸는 형태를 찾으려 한다.

row dominance에서는 minterm들을 담고 있는 집합이 있을 때, 포함되는 쪽 집합을 가진 pi를 제거한다. 11-1->=-1-10이면 pi 11-1-이 pi -1-10를 포함하고 있다는 것이다. 따라서 이 경우엔 -1-10를 second epi가 될 수 있는 후보들을 담고 있는 pi array에서 제외시킨다.

```

findEPI:
      9  13  15  17  25  26  27  31
0-1-- -  0  0  -  -  -  -  -
0--0-  0  0  -  -  -  -  -
110-1 -  -  -  -  0  -  0  -
11-1- -  -  -  -  -  0  0  0
--001  0  -  -  0  0  -  -  -
--11- -  -  0  -  -  -  -  0

epi: ['--001', '11-1-']
after del epi minterms: 13, 15,
nepi: ['0-1--', '0--0-', '110-1', '--11-']
-----
column dominance start:
pi: ['0-1--', '0--0-', '110-1', '--11-']
minterms: 13, 15,
      0-1--  0--0-  110-1  --11-
13  0      0      -      -
15  0      -      -      0

after column dominance minterms: 13, 15,
-----
row dominance:
      13  15
0-1--  0  0
0--0-  0  -
110-1 -  -
--11- -  0

```

1, 2, 3의 과정은 재귀 호출을 통해서 계속 진행된다.

minterms의 원소의 개수가 0이거나, 1, 2, 3의 과정을 거쳐도 minterms의 개수가 변하지 않는다면 (더 이상 표 size를 줄일 수 없다면) 루프를 벗어난다.

```
ret: ['0-1--', '0---0', '11-1-', '-01--', '--001']  
-----end solve-----}
```

이렇게 루프를 벗어나면 최종적으로 최적화를 위해 사용할 수 있는 epi와 secondary epi들을 리턴한다.