

CD, RD 구현 Report

논리회로 설계

제출일: 2022 05 27

목차



구현 코드 특징



Row Dominance



Column Dominance



Loop Break 조건, PM 여부

● 유사 딕셔너리 사용 - defaultdict(list)

```
PI: defaultdict(<class 'list'>, {'1020': [8], '1211': [11, 15], '1102': [12]})  
New Selected PI: ['1020', '1102', '1211']  
NEPI: defaultdict(<class 'list'>, {})
```

PI

minterms

Dict의 Key 값으로 PI를 설정
Key의 Value들로 minterm들을 list로 설정

과제제출: QM Finding EPI

1

1.0 (100%)

유사 딕셔너리 기반
PI, EPI 구현 문제 없었고
잘 해결되었습니다

Row Dominance

Row Dominance 적용

```
def applyRD(PI) :
```

```
    resultPI = PI.copy()
```

```
    for i in PI :
```

```
        for l in PI :
```

```
            if(checkListInList(PI[i], PI[l])==1 and i != l) :
```

```
                if(1 in resultPI) :
```

```
                    resultPI.pop(i)
```

```
    return resultPI
```

PI가 포괄하는 Minterm List가
다른 List를 지배하는지 확인



지배 받는 PI를 삭제

Row Dominance

Row Dominance

Before RD: defaultdict(<class 'list'>, {'1211': [11, 15], '1020': [8], '1012': [11], '1102': [12], '1121': [15]})

	8	11	12	15
1-11		0		0
10-0	0			
101-		0		
110-			0	
11-1				0

After RD: defaultdict(<class 'list'>, {'1211': [11, 15], '1020': [8], '1102': [12]})

Before RD(PI): ['101-', '10-0', '110-', '11-1', '1-11']

After RD(PI): ['10-0', '110-', '1-11']

Column Dominance

```
CD: defaultdict(<class 'list'>, {'1211': [11, 15], '1102': [12], '1012': [11], '1020': [8], '1121': [15]})
```

```
CD reversed: defaultdict(<class 'list'>, {8: ['1020'], 11: ['1012', '1211'], 12: ['1102'], 15: ['1211', '1121']})
```

Column Dominance 적용을 더 쉽게 하기 위해서
Dict의 **Key(PI)**와 **Value(Minterm)**를 서로 전환

	8	11	12	15
1-11				
10-0				
101-				
110-				
11-1				



	1-11	10-0	101-	110-	11-1
8					
11					
12					
15					

Column Dominance

Column Dominance 적용

```
def applyCD(PI) :
```

```
#Dict의 Key를 PI에서 Minterm으로 전환
```

```
reversedMT = reversePIandMT(PI)
```

Key와 Value 전환

```
resultMT = reversedMT.copy()
```

```
print("Before CD reversed: ", resultMT)
```

```
for i in reversedMT :
```

```
    for l in reversedMT :
```

```
        # 특정 Minterm을 cover하는 PIO이 다른 Minterm을 cover하는 PI에 포함 여부 확인
```

```
        if(checkListInList(reversedMT[i], reversedMT[l])==1 and i != l) :
```

```
            if(l in resultMT) :
```

```
                resultMT.pop(l)
```

지배 하는 MT를 삭제

```
print("After CD reversed: ", resultMT)
```

```
return reversePIandMT(resultMT)
```

다시 전환된 Dict를 리턴

Column Dominance

Column Dominance

Before CD reversed: defaultdict(<class 'list'>, {8: ['1020'], 10: ['1020', '1012'], 11: ['1012', '1211'], 12: ['1102'], 13: ['1102', '1121'], 15: ['1211', '1121']})
After CD reversed: defaultdict(<class 'list'>, {8: ['1020'], 11: ['1012', '1211'], 12: ['1102'], 15: ['1211', '1121']})

	8	10	11	12	13	15
10-0	O	O				
101-		O	O			
1-11			O			O
110-				O	O	
11-1					O	O

After CD: defaultdict(<class 'list'>, {'1211': [11, 15], '1102': [12], '1012': [11], '1020': [8], '1121': [15]})

Before CD(minterm): [8, 10, 11, 12, 13, 15]
After CD(minterm): [8, 11, 12, 15]

Step 3: Minimum Cover

- Summary
 - (1) Find all PIs to construct a PI table
 - (2) Find EPIs to simplify the table
 - No NEPI remained? Then quit.
 - (3) Apply column dominance row
 - (4) Apply row dominance row
 - (5) Any simplification made from (2) and (3)?
 - Yes: go to (2)
 - No: move on to the next slide to learn Petrick's method

(1) Find all PIs to construct a PI table

```
while(True) :  
    # 무한 루프 시작  
    print("LOOP");print()  
    SelectedPI = PNE.MakeEPI(PI)  
  
    # 처음 실행할 때 Selected PI는 EPI  
    if(checkFirst == 0) :  
        EPI = SelectedPI ; checkFirst = 1 ;  
        print("2) EPI 추출 : ")  
        print("EPI: ",sortAndReplace(EPI))  
    else :  
        AllSelectedPI.extend(SelectedPI)  
        print("2) Table 단순화 후(CD, RD 적용) 뽑아낸 EPI: ")  
        print("New Selected PI: ",sortAndReplace(SelectedPI))  
    print()  
  
    # NEPI PI에서 Selected PI 제거  
    NEPI = PNE.removeEPIonPI(PI,SelectedPI)  
    print("아직 선택되지 않은 PI 추출(NEPI or NEPI After CD, RD)")  
    print("NEPI: ",sortAndReplace(list(NEPI.keys())))  
    print()  
  
    # NEPI가 없다면 종료  
    if(len(NEPI) == 0) :  
        NeedPM = False ;  
        print("2-1) NEPI없음 break"); break
```

(2) Find EPIs to simplify the table

- No NEPI remained? Then quit.

```
# NEPI가 없다면 종료  
if(len(NEPI) == 0) :  
    NeedPM = False ;  
    print("2-1) NEPI없음 break"); break
```

Loop Break / PM 여부

```
# Column Dominance 적용
print("3) Column Dominance 적용");print()
#print("Before CD(dict): ",dict(NEPI)) # Dict의 내용물이 필요하다면 주석해제
print("Before CD(minterm): ",sortMinterm(NEPI.values()))
NEPI = Dom.applyCD(NEPI)
#print("After CD(dict): ",dict(NEPI)) # Dict의 내용물이 필요하다면 주석해제
print("After CD(minterm): ",sortMinterm(NEPI.values()))
```

(3) Apply column dominance raw

```
# Row Dominance 적용
print(); print("4) Row Dominance 적용"); print()
#print("Before RD(dict): ",dict(NEPI)) # Dict의 내용물이 필요하다면 주석해제
print("Before RD(PI): ",sortAndReplace(dict(NEPI)))
NEPI = Dom.applyRD(NEPI)
#print("After RD(dict): ",dict(NEPI)) # Dict의 내용물이 필요하다면 주석해제
print("After RD(PI): ",sortAndReplace(dict(NEPI)))
```

(4) Apply row dominance raw

```
# CD, RD 적용 후에도 PI 변화 없을 시 종료
print()
if(len(PI) == len(NEPI) and len(Dom.reversePIandMT(PI)) == len(Dom.reversePIandMT(NEPI))) :
    print("5-2) CD와 RD 적용해도 변화 없음 PM 필요"); break
else : print("5-1) 2번으로 되돌아감")
PI = NEPI
```

(5) Any simplification made from (2) and (3)?

- Yes: go to (2)
- No: move on to the next slide to learn Petrick's method

Loop Break / PM 여부

```
# NEPI가 남아있고 CD와 RD 둘 다 적용 불가능 할 경우  
print()  
if(NeedPM == True) : print("Need Petrick's Method")  
else : print("Finish(Not need Petrick's Method)")
```

(2) Find EPIs to simplify the table

- No NEPI remained? Then quit.

(5) Any simplification made from (2) and (3)?

- Yes: go to (2)
- No: move on to the next slide to learn Petrick's method

Test Case Run

● minterm = [4, 8, 0, 4, 8, 10, 11, 12, 13, 15]

1) PI추출

PI : ['101-', '10-0', '110-', '11-1', '1-11', '--00']

LOOP

2) EPI 추출 :

EPI: ['--00']

아직 선택되지 않은 PI 추출(NEPI or NEPI After CD, RD)

NEPI: ['101-', '10-0', '110-', '11-1', '1-11']

3) Column Dominance 적용

Before CD(minterm): [8, 10, 11, 12, 13, 15]

After CD(minterm): [8, 11, 12, 15]

4) Row Dominance 적용

Before RD(PI): ['101-', '10-0', '110-', '11-1', '1-11']

After RD(PI): ['10-0', '110-', '1-11']

5-1) 2번으로 되돌아감

LOOP

2) Table 단순화 후(CD, RD 적용) 뽑아낸 EPI:

New Selected PI: ['10-0', '110-', '1-11']

아직 선택되지 않은 PI 추출(NEPI or NEPI After CD, RD)

NEPI: []

2-1) NEPI없음 break

Finish(Not need Petrick's Method)

Test Case Run

minterm = [4, 8, 0, 4, 8, 10, 11, 12, 13, 15]

Prime implicants	Minterm							
	0	4	8	10	11	12	13	15
P1 = 1 0 - 0			v	v				
P2 = 1 0 1 -				v	v			
P3 = 1 1 0 -						v	v	
P4 = 1 - 1 1					v			v
P5 = 1 1 - 1							v	v
P6 = - - 0 0	v	v	v			v		

Test Case Run



minterm = [3, 6, 0, 1, 2, 5, 6, 7]

```
1) PI추출
PI : ['00-', '0-0', '11-', '1-1', '-01', '-10']

LOOP

2) EPI 추출 :
EPI: []

마직 선택되지 않은 PI 추출(NEPI or NEPI After CD, RD)
NEPI: ['00-', '0-0', '11-', '1-1', '-01', '-10']

3) Column Dominance 적용

Before CD(minterm): [0, 1, 2, 5, 6, 7]
After CD(minterm): [0, 1, 2, 5, 6, 7]

4) Row Dominance 적용

Before RD(PI): ['00-', '0-0', '11-', '1-1', '-01', '-10']
After RD(PI): ['00-', '0-0', '11-', '1-1', '-01', '-10']

5-2) CD와 RD 적용해도 변화 없음 PM 필요

Need Petrick's Method
```


Test Case Run

minterm = [4, 8, 0, 4, 8, 10, 11, 12, 13, 15]

Prime implicants	Minterm						
	0	1	2	5	6	7	
P1 = 0 0 - (0,1) $a'b'$	v	v					
P2 = 0 - 0 (0,2) $a'c'$	v		v				
P3 = - 0 1 (1,5) $b'c$		v		v			
P4 = - 1 0 (2,6) bc'			v		v		
P5 = 1 - 1 (5,7) ac				v		v	
P6 = 1 1 - (6,7) ab					v	v	

감사합니다

Reference



**국민대학교 논리회로 설계 lecture5_tabular_method PDF
(8, 12, 16 p)**