LECTURE 5 Monday

1. Web Page
2. Homework - project presentation (start Monday?)
3. Book
4. Scan pages

MMD = Miller, Maslov, Dueck

Group 1. MMD with don't cares

Group 2 MMD with Miller's UVs

Group 3 Comparison of linearly Independent circuits

Group 4 Toolkit in Matlab for CC → coming Monday

I spent first part of lecture to introduce these projects

free Matlab clone ~95% the same

"Scilab"

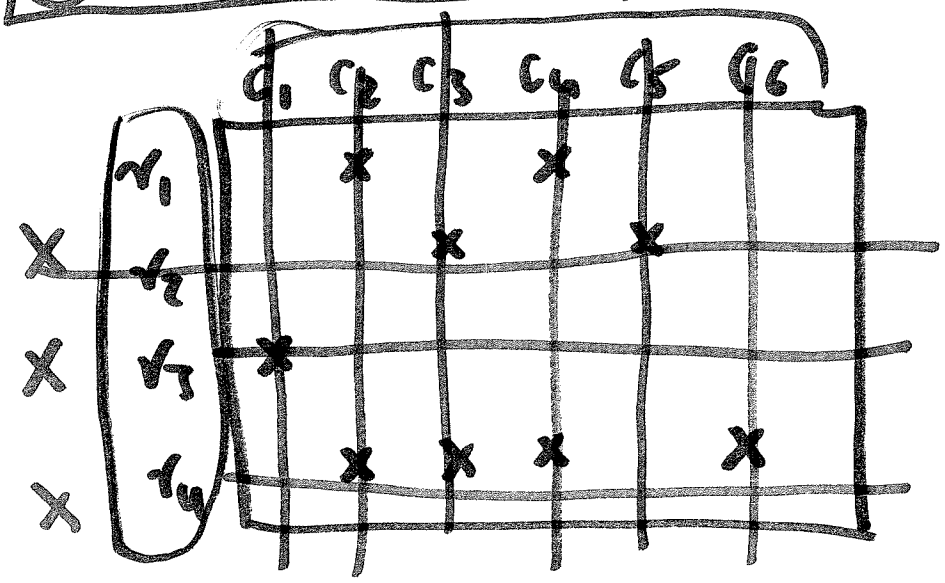
QuiddPro

Viamontes

Univ. of Michigan

1. Unate covering
2. Binate covering
3. Even-odd covering
4. SAT satisfiability
5. Maximum clique
6. Graph coloring

① Unate covering



Unate covering

↓ Reduce

Petrick function Minimization

the idea of problem reduction

$$r_3 \cdot (r_1 + r_4) \cdot (r_2 + r_4) \cdot (r_1 + r_4)$$

$$\cdot r_2 \cdot r_4 = 1$$

$\downarrow r_4 = 1$

$r_3 \cdot r_2$

r_4, r_3, r_2

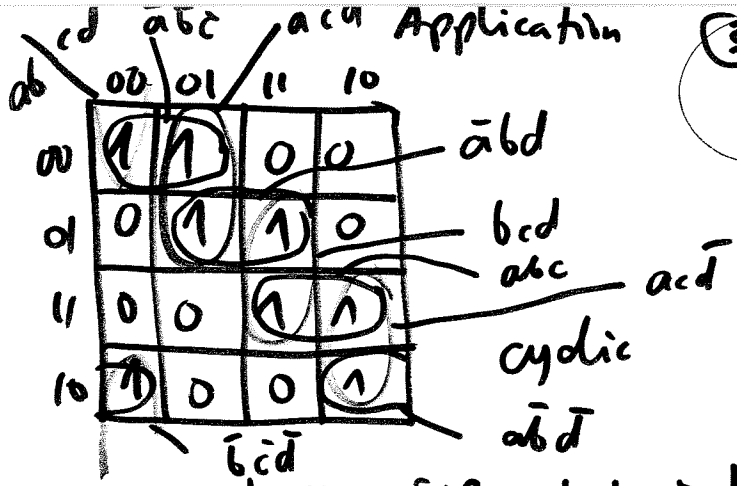
$\downarrow r_3 = 1$

$r_2 = 1$

1

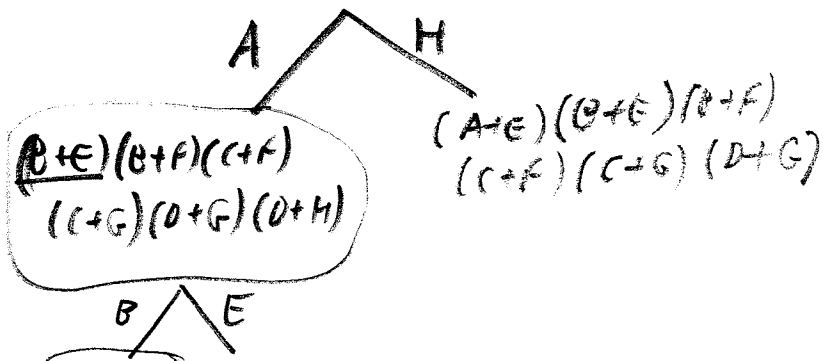
no search

COVERING TABLE AND PETRICK FUNCTION



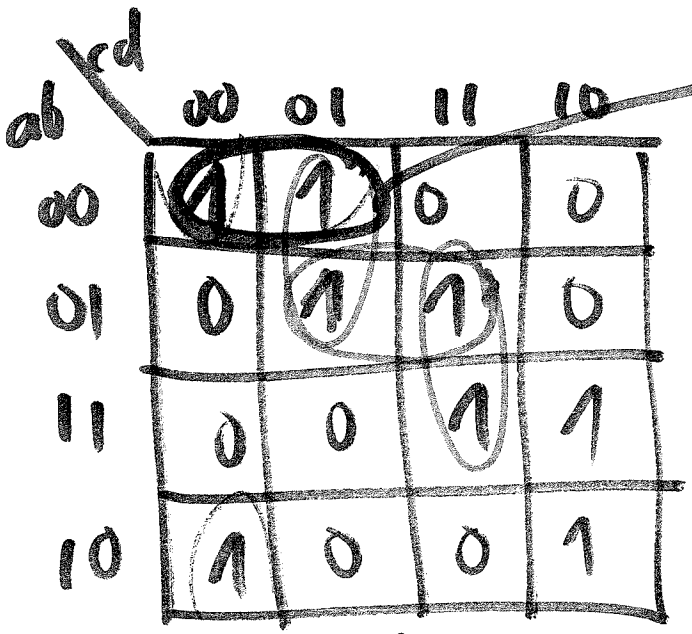
We convert the SOP minimization problem to unate covering. POS minimization is solved the same way

$$1 = (A+H)(A+E)(B+E)(B+F)(C+F)(C+G)(D+G)(D+H)$$



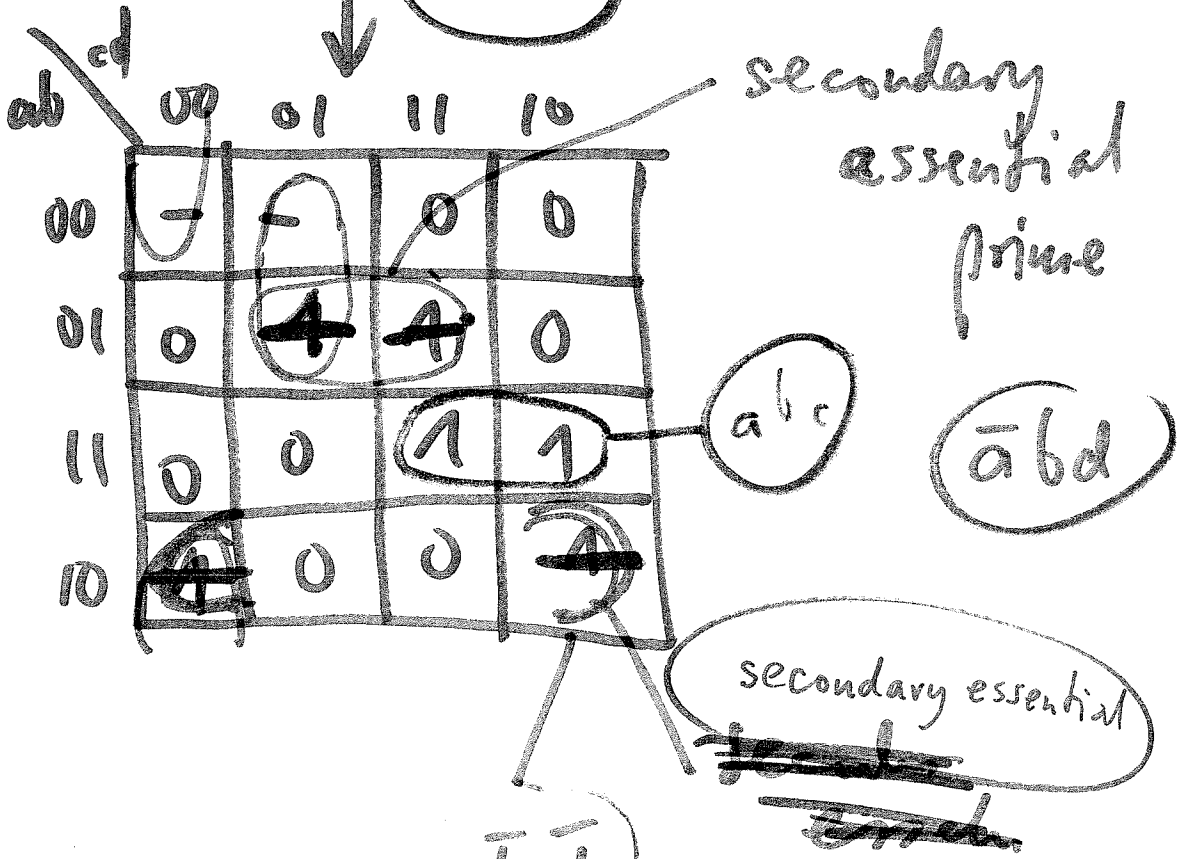
		0000	0001	0101	0111	1111	1110	1010	1000
A	$\bar{a}\bar{b}\bar{c}$ = 000X	x	x						
B	$\bar{a}bd$ = 01X1			x	x				
C	abc = 111X					x	x		
D	$a\bar{b}\bar{d}$ = 10X0							x	x
E	$\bar{a}\bar{c}d$ = 0X01		x	x					
F	bcd = X111				x	x			
G	$acd\bar{b}$ = 1X10						x	x	
H	$\bar{b}\bar{c}\bar{d}$ = X000	x							x

I found solution ABCD of cost 4 terms



I select (randomly this group)

$\bar{a}b\bar{c}$ ← not essential



secondary essential prime

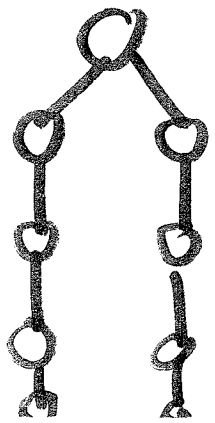
abc

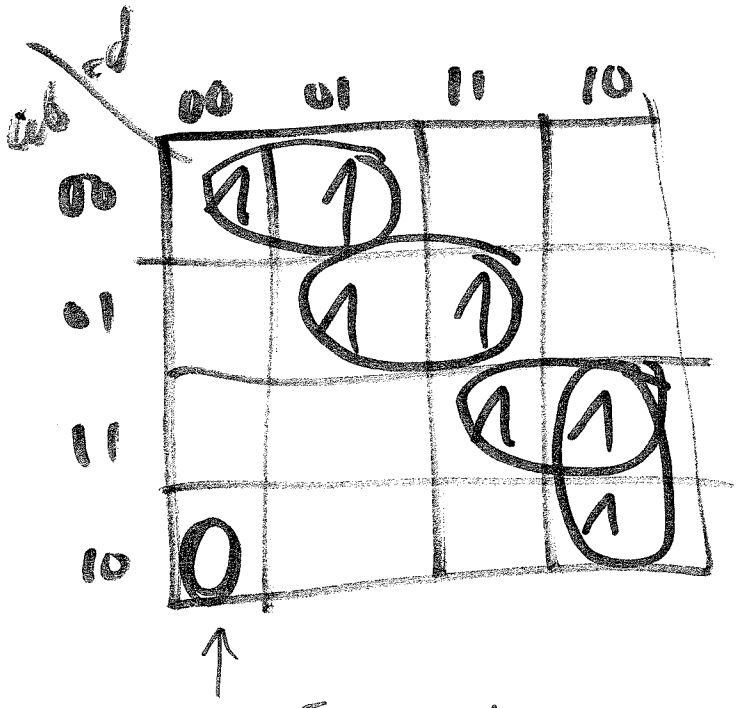
$\bar{a}bd$

secondary essential

$ab\bar{d}$

Here I illustrated how using secondary essential primes simplifies the process of finding exact solutions

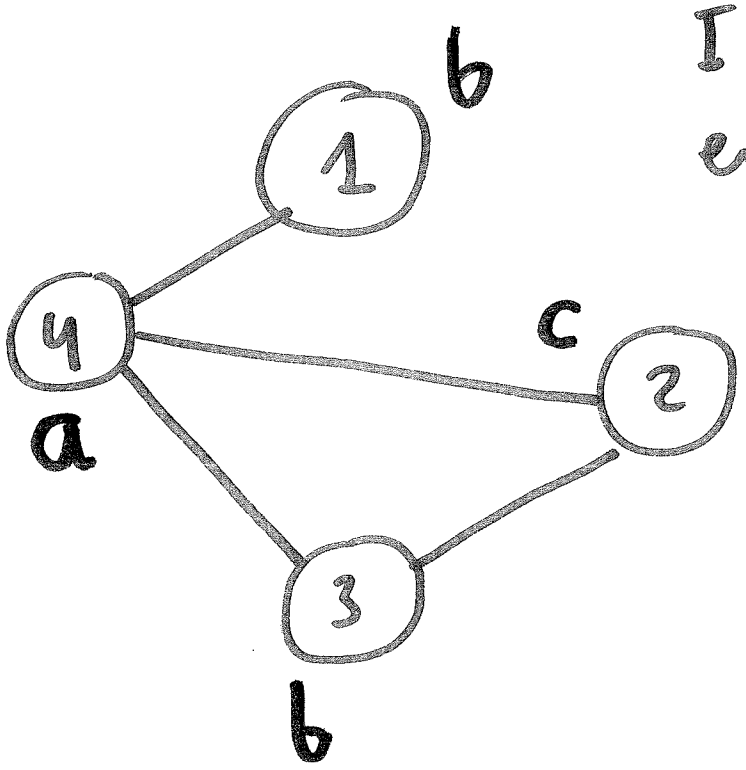




In this case the solution is found without branching.
EXACT!

↑
now I do only one change to Kmap.
My function has a 0 here.
Now I can find solution
Using secondary essential primes and I have exact solution without branching!!

Graph coloring



I want to color every node with one color.

Any two adjacent nodes should have different colors

We want to find coloring with minimum number of colors.

1 found coloring

1 — b — a
2 — c — b
3 — b — a
4 — a — c

↑ proper graph coloring

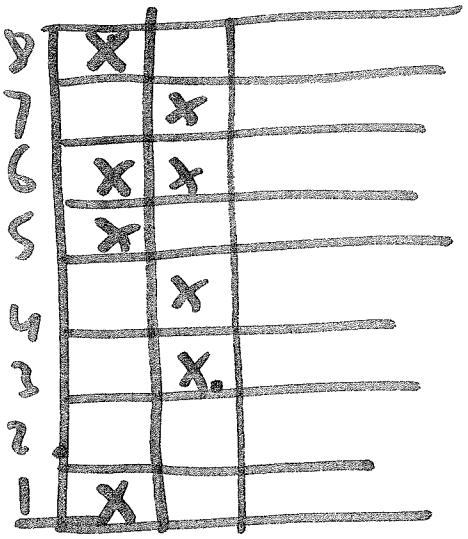
↑ Chromatic number.

another equivalent solution. (not interesting)

Graph coloring has many applications

FUNNY EXAMPLE

HOUSE WITH 8 elevators



every elevator can connect not more than 4 elevators

Question

How many elevators we need that by selecting elevator one can go directly from every elevator to every other elevator?

Solve using Petrick function.
First draw the covering table.
Think how to use symmetry (cyclicity) of the covering table?!

BINATE COVERING

$(a+b+c)$ such terms are in unate covering. We use them also in binate covering

We have such terms in binate covering:

$$(\bar{a} \rightarrow b+c) \equiv (a+b+c)$$

$$(d \rightarrow b+e) \equiv (\bar{d}+b+e)$$

$$(\bar{d}e \rightarrow b+f) \equiv (\overline{\bar{d}e}+b+f) = (de+b+f)$$

Implication $a \rightarrow b$

a	b	$a \rightarrow b$	$\bar{a}+b$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

↑ here we checked the definition of implication operator

$$(a \rightarrow b) \equiv (\bar{a}+b)$$

↑ this is meta operator

$$(A + BE)$$
$$= (A + B)\cdot(A + E)$$
$$= A\cdot A + AE + BA + BE$$
$$= A(1 + E + B) + BE$$
$$= A + BE$$

After all rule-based preprocessing such as implication removal, de Morgan and we get the bitote covering as a SAT formula

$$(a + \bar{b} + \bar{c})(\bar{a} + c + \bar{d}) \cdots = 1$$

Use of implications

$A \to F$

$A \to G$

$A \to H$

$(\bar{A} + F) \ldots (\bar{A} + G) \cdots (\bar{A} + H)$

$A = 1$

$F \ldots G \cdots H$

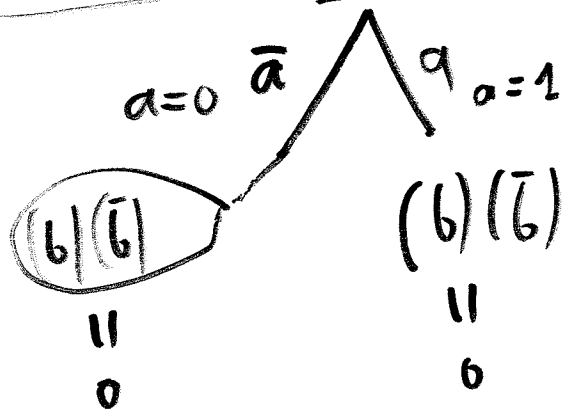
F, G, H

continue search to terminate

$(a+b)(a+\bar{b})(\bar{a}+b)(\bar{a}+\bar{b})(\quad)$ SAT

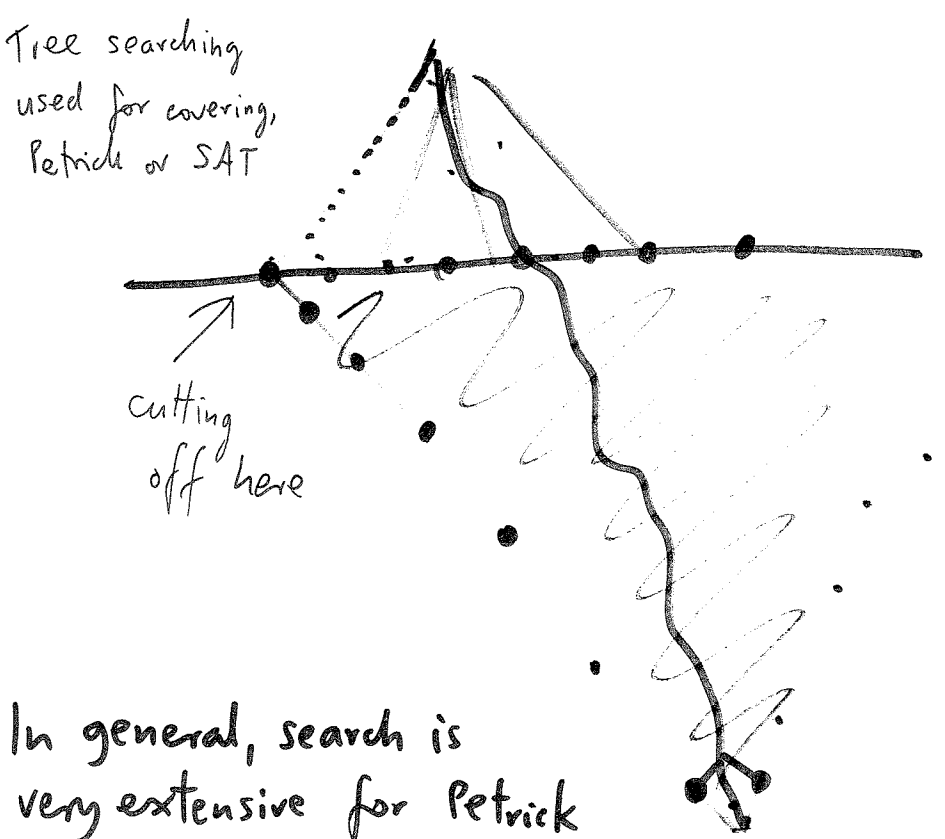
$(x+y+z)$

I can have complex formula using these and other variables here



Example of formula that is not satisfied

This is a kmap with only zeros!



In general, search is very extensive for Petrick Function.

This is similar to binate covering and SAT

SAT = SATisfiability (of formula)