Quine-McCluskey

1- **Analysis phase** - Prime Implicants search & extraction

a. Visual enumeration of ALL sub-cubes (problematic if > 5 variables)

b. Tabular method

i. Build the associated Truth Table

ii. Group minterms according to # of 'ones' in each.

Groups are named G_i where i = # 'ones, they are sorted in ascending order.

Minimum is 0 groups => F=0, maximum is n+1 groups.

iii. Compare Hamming distance, for each minterm of G_i and G_{i+1}.

If H.d.=1, mark & group those minterms in G_i' using don't cares.

If minterm is not used, it's a Prime Implicant.

iv. Repeat *ii, iii* to build G_i'', G_i''', etc. until cannot group anymore.

Be careful! Don't cares are also considered for the H.d. ~~−000~~ ~~01−0~~

v. Remaining minterms cannot be used, so they are Prime Implicants.

Keep only one PI if, for example:

G_0''	−0−	(0,1,4,5)	PI1
	−0−	(0,4,1,5)	

If incompletely specified logic functions,

$f(a,b,c)=\sum m(0,1,3,4,5,7)+\sum d(2,6)$

=> hypothesis that all don't cares are 1 (as they were in $\sum$m)

2- **Synthesis phase** - Minimal cover of the logic function

a. Minimal function cover table (graphical method, not very good)

i. Build a table from minterms -without don't cares- (lines) and found Prime Implicants.

ii. For each PI mark all minterms covered by this PI.

iii. Find Essential PIs (the only ones covering certain minterms)

iv. Find PIs so that all minterms are covered.

b. Minimal cover equation (Algebraic method that generates all possible solutions and that can be automatized)

i. For each minterm: OR all PI covering this minterm.

So that if i_1, i_2 are in the same line, we obtain (i_1+i_2)

ii. We AND all results so that if (i_1+i_2), (i_1+i_5) covers different lines:

we have $(i_1+i_2)\cdot(i_1+i_5)$. Essential PIs will be isolated terms.

iii. Drop double terms and terms containing an EPI.

iv. Distribute terms like that: (i_1+i_4) (i_1+i_2) -> $(i_1+i_2\cdot i_4)$ and make sure there is no parenthesis anymore.

The result is an OR of possible solutions (a SoP of PI).

v. Choose the solution with the least number of smallest possible gates and rewrite it as an OR of PI: $i_2\cdot i_4$ -> PI_2 + PI_4