Configurational Analysis beyond the Quine-McCluskey algorithm

Adrian Dușa

University of Bucharest

Milan, 8 February 2018

QCA and the Boolean minimization process

"If two or more instances of the phenomenon under investigation have only one circumstance in common, the circumstance in which alone all the instances agree, is the cause (or effect) of the given phenomenon."

John Stuart Mill, A System of Logic, Vol. 1. 1843. p. 454.

A	B	Y
1	1	1
1	0	1
1	–	1

Boolean minimization algorithms

- classical Quine-McCluskey (QMC)
- improved QMC
- enhanced QMC (eQMC)
- Espresso
- CCubes

All of these algorithms have a truth table as an input.

The truth table

	A	B	C	D
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

These are all possible (16) configurations of 4 causal conditions, where the total number of rows is usually equal to 2^k, where k is the number of causal conditions.

In fact, the real number of rows for a truth table is:

$$TT = \prod_{c=1}^{k} l_c \tag{1}$$

The truth table

	A	B	C	D	OUT
1	0	0	0	0	
2	0	0	0	1	
3	0	0	1	0	
4	0	0	1	1	
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	
8	0	1	1	1	
9	1	0	0	0	1
10	1	0	0	1	
11	1	0	1	0	1
12	1	0	1	1	
13	1	1	0	0	
14	1	1	0	1	0
15	1	1	1	0	
16	1	1	1	1	0

These are all possible (16) configurations of 4 causal conditions, where the total number of rows is usually equal to 2^k, where k is the number of causal conditions.

In fact, the real number of rows for a truth table is:

$$TT = \prod_{c=1}^{k} l_c$$

The truth table

	A	B	C	D	OUT
1	0	0	0	0	?
2	0	0	0	1	?
3	0	0	1	0	?
4	0	0	1	1	?
5	0	1	0	0	0
6	0	1	0	1	1
7	0	1	1	0	?
8	0	1	1	1	?
9	1	0	0	0	1
10	1	0	0	1	?
11	1	0	1	0	1
12	1	0	1	1	?
13	1	1	0	0	?
14	1	1	0	1	0
15	1	1	1	0	?
16	1	1	1	1	0

Main solution types:

- Conservative (complex) solution: minimization based on the observed positive output configurations only
- Parsimonious solution: the observed positive output configurations plus the remainders (aka “don’t cares” in electrical engineering)

Classical QMC

It is a two-level logic minimization, with:

- Step 1. Find all prime implicants (PIs) of a given set of initial configurations (with or without including the remainders)
- Step 2. Solve the PI chart that translates the PIs to the positive output configurations

Classical QMC step 1: finding the prime implicants

- verify all possible pairs of rows
- for each pair of rows, check all pairs of columns for differences
- if there is a single difference, minimize the pair of rows
- otherwise return

This procedure is NP complete (nondeterministic polynomial time), with exponential increase in complexity and memory consumption.

Classical QMC step 2: solving the PI chart

	6	9	11
$\sim$B	–	x	x
$\sim$AD	x	–	–
A$\sim$D	–	x	x
C$\sim$D	–	–	x

Find all possible combinations of (a minimal number of) prime implicants that cover all initial positive output configurations.

Two models in the parsimonious solution:

1. $\sim$AD + $\sim$B
2. $\sim$AD + A$\sim$D

Improved (step 1 of) QMC: the $2x - y$ rule

	A	B	C	D
9	1	0	0	0
11	1	0	1	0
	1	0	–	0

Translated in the 3^k matrix:

	A	B	C	D
68	2	1	1	1
71	2	1	2	1
65	2	1	0	1

$$2 \cdot 68 - 71 = 65$$

Improved QMC

Dușa (2010) "A mathematical approach to the Boolean minimization problem", Quality & Quantity 44(1), 99-113

All possible prime implicants are found in the implicant matrix: for binary causal conditions, it has 3^k rows. A more general formula that is valid for multi-value causal conditions is:

$$IM = \prod_{c=1}^{k} (l_c + 1) \tag{2}$$

where l_c is the number of levels for condition $c = 1 \dots k$

Improved QMC

Further mathematical tricks are applied to determine which pairs of rows can be minimized, to make sure the $2x - y$ rule is applied only where possible, requiring either computation time or memory to store the matrix of minimizable pairs of rows (that has 2^k rows and k columns).

Compared to classical QCA that has an upper complexity ceiling of about 10-11 causal conditions, it managed to:

- increase the complexity to about 14-15 causal conditions (the difference between 3^{12} and 3^{15})
- find solutions in many orders of magnitude less time than QMC

eQMC

It only concentrates on the observed (positive and the negative) instances.

	A	B	C	D	OUT
6	0	1	0	1	1
9	1	0	0	0	1
11	1	0	1	0	1
5	0	1	0	0	0
14	1	1	0	1	0
16	1	1	1	1	0

	A	B	C	D
51	1	2	1	2
68	2	1	1	1
71	2	1	2	1
50	1	2	1	1
78	2	2	1	2
81	2	2	2	2

eQMC

Supersets of the positive output configurations:

```
 2  3  4  5  6  7  8 10 11 13 14 16 17 19 21 22 24 28 30 31
33 46 48 49 51 55 56 58 59 61 62 64 65 67 68 70 71
```

Supersets of the negative output configurations:

```
 2  3  4  5  6  7  9 19 20 21 22 23 24 25 27 28 29 31 32 46
47 49 50 55 57 58 60 61 63 73 75 76 78 79 81
```

Difference:

```
8 10 11 13 14 16 17 30 33 48 51 56 59 62 64 65 67 68 70 71
```

eQMC

	A	B	C	D
8	0	0	2	1
10	0	1	0	0
11	0	1	0	1
13	0	1	1	0
14	0	1	1	1
16	0	1	2	0
17	0	1	2	1
30	1	0	0	2
33	1	0	1	2
48	1	2	0	2
51	1	2	1	2
56	2	0	0	1
59	2	0	1	1
62	2	0	2	1
64	2	1	0	0
65	2	1	0	1
67	2	1	1	0
68	2	1	1	1
70	2	1	2	0
71	2	1	2	1

Non-redundant PIs:

8 10 30 56

	A	B	C	D	
8	0	0	2	1	C∼D
10	0	1	0	0	∼B
30	1	0	0	2	∼AD
56	2	0	0	1	A∼D

eQMC

Dușa and Thiem (2015) "Enhancing the Minimization of Boolean and Multivalue Output Functions With eQMC", Journal of Mathematical Sociology 39, 92-108

The complexity level reached 18 causal conditions, with:

- a minimal memory consumption, decreasing from about 1.5 GB for the improved QMC, to some tens of MB for eQMC
- hundreds of times faster than the (improved) QMC

Espresso

- Developed more than 30 years ago at MIT
- Different method from QMC: instead of expanding a logic function with minterms, it iteratively manipulates the product terms in the ON, OFF and DC sets
- Can deal with any number of inputs and outputs
- It is a heuristic logic minimizer, with a very quick solution that is not guaranteed to be a global minimum (just very closely approximated)
- Not suitable for QCA, where we need all possible solutions for a given problem, not just the first quick one

CCubes

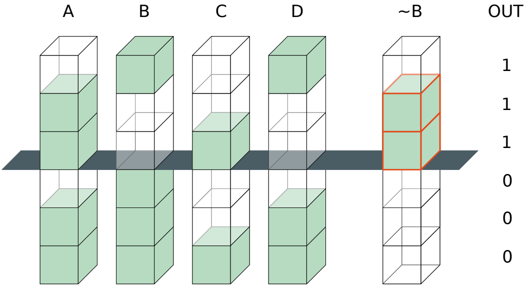
Top - down
From complex towards parsimonious

Bottom - up
From parsimonious towards more complex

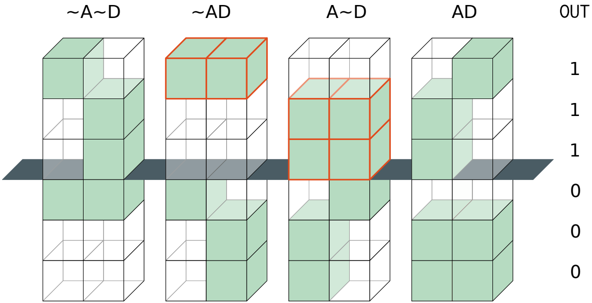
Definition: A prime implicant is the simplest possible, non-redundant, fully consistent superset of a positive output configuration.

CCubes

A	B	C	D
0	1	0	1
1	0	0	0
1	0	1	0
0	1	0	0
1	1	0	1
1	1	1	1



CCubes



CCubes: partitioning the search space

The implicant matrix from equation 2 can be partitioned to:

$$S_{MV} = \sum_{c=1}^{k} \binom{k}{c} \prod_{s=1}^{c} l_s \tag{3}$$

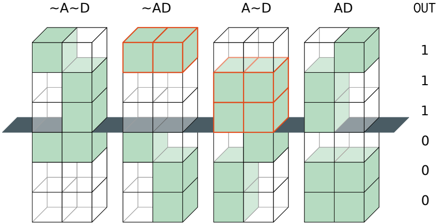
$$S_{CS} = \sum_{c=1}^{k} \binom{k}{c} \prod_{s=1}^{c} 2 = \sum_{c=1}^{k} \binom{k}{c} 2^c = 3^k - 1 \tag{4}$$

CCubes: partitioning the search space

The following equality (between equations 2 and 3) holds:

$$\prod_{c=1}^{k}(l_c + 1) \;=\; 1 + \sum_{c=1}^{k}\binom{k}{c}\prod_{s=1}^{c} l_s$$

CCubes: shortening the search space



A	D	OUT
0	1	1
1	0	1
1	0	1
0	0	0
1	1	0
1	1	0

- Not all combinations of 2 out of 4 conditions are meaningful / useful
- Concentrate *only* on the observed combinations, which are already found in the truth table
- Both ∼AD and A∼D are fully consistent with the positive output, because they are not found in the negative output configurations.

CCubes: search depth

Usually, about half of the entire search space is void, does not produce any prime implicants.

$$\sum_{c=1}^{k} \binom{k}{c} \prod_{s=1}^{c} l_s = P + U + N \tag{5}$$

where:

P = prime implicants

U = unique subsets pf the prime implicants

N = non prime implicants (supersets of the negative configurations)

CCubes: search depth and `min.pin`

Check after each complexity level if:

- the PI chart **can** be solved
- the PI chart can be minimally solved: the disjunctive solution has the same length as the one from the previous level of complexity
- the PIs generated at the current complexity level contribute to minimally solving the PI chart (argument `min.pin`)

Performance benchmark

- hundreds of times faster than eQMC
- works with up to 30 conditions (compared to a maximum of 18 for eQMC)
- the Espresso challenge: 1000 random samples, 20 conditions and 20 cases

Espresso
$\bar{x} = 2.017$ seconds
$s = 1.277$ seconds

CCubes
$\bar{x} = 0.941$ seconds
$s = 0.550$ seconds
$\bar{x}_{MP} = 0.571$ seconds

Dușa (2018, forthcoming) "Consistency Cubes: A Fast, Efficient Method for exact Boolean Minimization.", R Journal.

CCubes: pseudo-counterfactual method

- CCubes does not explicitly uses the remainders
- but it always finds exactly the same solutions as the classical QMC when including the remainders
- it can be used to find intermediate solutions
- it can be used for ESA, by excluding the problematic remainders (which are added to the set of negative output configurations)

Thank you

Dușa (2018, forthcoming) *QCA with R. A comprehensive resource.*
https://bookdown.org/dusadrian/QCAbook/

dusa.adrian@unibuc.ro