Notes: Complex boolean functions

Carolin Zöbelein*

July 15, 2020

DRAFT

Abstract

We examine the boolean functions `NEG`, `AND` and `OR` on the field of complex numbers $\mathbb{C}$ with boolean values $\{i, 1\}$ and the map between them and the classical definition with $\{0, 1\}$.

Keywords: XXX, YYY, ZZZ
2010 Mathematics Classification: XXX, YYY, ZZZ

Preamble

The following content is a sketch for discussion purposes only, without warranty for mathematical completeness.

1 Introduction

In this short notes, we will do an analysis of boolean functions on the field of complex numbers $\mathbb{C}$ for a specific defined case regarding the values *True* and *False*, based on the work of Ryan ODonnell [1], Zilong Wang and Guang Gong [2] and Zhiwei Zhang [3] [4].

2 Preview

Now, a short preview of the final results. Given are integer boolean values $x \in \{0, 1\}$ and complex values $z \in \mathbb{C}$. For the boolean cases `NEG`, `AND` and `OR` we will use the two complex boolean values $\{i, 1\}$.

We will denote the inverse boolean value of x by $\bar{x}$ respectively for z by $\bar{z}$. For the complex conjungate of z, we will write z^c.

In table 1, you can see the values for `NEG` of x and z, which gives us several statements.

Theorem 2.1 (`NEG` of x)**.** *The inverse boolean value of x, generated by `NEG`(x), is given by $\bar{x} = 1 - x$.*

Theorem 2.2 (`NEG` of z)**.** *The inverse boolean value of z, generated by `NEG`(z), is given by $\bar{z} = (1 + i) - z$ and by $\bar{z} = z^3 i$.*

For `AND` and `OR` we need an helper boolean function, called *generator*.

*The author believes in the importance of the independence of research and is funded by the public community. If you also believe in this values, you can find ways for supporting the author's work here: `https://research.carolin-zoebelein.de/funding.html`, Email: `contact@carolin-zoebelein.de`, PGP: D4A7 35E8 D47F 801F 2CF6 2BA7 927A FD3C DE47 E13B, `https://research.carolin-zoebelein.de`

Table 1: NEG

x	$-x$	$\bar{x}$	z	$-z$	$\bar{z}$
0	0	1	i	-i	1
1	-1	0	1	- 1	i

Theorem 2.3 (*Generator* g of (x_i, x_j)). *The generator* g *of* (x_i, x_j)*, which maps* $(0,0) \mapsto 1$*,* $(0,1) \mapsto 0$*,* $(1,0) \mapsto 0$ *and* $(1,1) \mapsto 0$*, is given by*

$$\begin{aligned} g(x_i, x_j) &= \left((x_i x_j + (x_i + x_j) - 1) \frac{1}{2} - x_i x_j \right) (-2) \\ &= -x_i x_j + x_i + x_j - 1. \end{aligned} \tag{1}$$

Theorem 2.4 (*Generator* g of (z_i, z_j)). *The generator* g *of* (z_i, z_j)*, which maps* $(i,i) \mapsto i$*,* $(i,1) \mapsto 0$*,* $(1,i) \mapsto 0$ *and* $(1,1) \mapsto 0$*, is given by*

$$\begin{aligned} g(z_i, z_j) &= (z_i z_j + (z_i + z_j) - 1) \frac{1}{2} - z_i z_j \\ &= (-z_i z_j + z_i + z_j - 1) \frac{1}{2}. \end{aligned} \tag{2}$$

In table 2, you can see the values for `AND` of (x_i, x_j) and (z_i, z_j).

Table 2: AND

x_i	x_j	$x_i x_j$	$x_i \wedge x_j$	z_i	z_j	$z_i z_j$	$z_i \wedge z_j$
0	0	0	0	i	i	-1	i
0	1	0	0	i	1	i	i
1	0	0	0	1	i	i	i
1	1	1	1	1	1	1	1

Theorem 2.5 (`AND` of (x_i, x_j)). *The boolean value of* $x_i \wedge x_j$*, generated by* `AND`(x_i, x_j)*, is given by*

$$\begin{aligned} x_i \wedge x_j &= x_i x_j \\ &= g(\bar{x}_i, \bar{x}_j). \end{aligned} \tag{3}$$

Theorem 2.6 (`AND` of (z_i, z_j)). *The boolean value of* $z_i \wedge z_j$*, generated by* `AND`(z_i, z_j)*, is given by*

$$\begin{aligned} z_i \wedge z_j &= z_i z_j + (1 - i) g(z_i, z_j) \\ &= i - (1 + i) g(\bar{z}_i, \bar{z}_j). \end{aligned} \tag{4}$$

Finally, in table 3, you can see the values for `OR` of (x_i, x_j) and (z_i, z_j).

Theorem 2.7 (`OR` of (x_i, x_j)). *The boolean value of* $x_i \vee x_j$*, generated by* `OR`(x_i, x_j)*, is given by*

$$\begin{aligned} x_i \vee x_j &= (x_i + x_j) - g(\bar{x}_i, \bar{x}_j) \\ &= 1 - g(x_i, x_j). \end{aligned} \tag{5}$$

Table 3: OR

x_i	x_j	$x_i + x_j$	$x_i \vee x_j$	z_i	z_j	$z_i + z_j$	$z_i \vee z_j$
0	0	0	0	i	i	2i	i
0	1	1	1	i	1	1 + i	1
1	0	1	1	1	i	1 + i	1
1	1	2	1	1	1	2	1

Theorem 2.8 (`OR` of (z_i, z_j))**.** *The boolean value of $z_i \vee z_j$, generated by `OR`(z_i, z_j), is given by*

$$\begin{aligned} z_i \vee z_j &= (z_i + z_j) - i + (1+i)\, g\left(\bar{z}_i, \bar{z}_j\right) \\ &= 1 + (1+i)\, g\left(z_i, z_j\right). \end{aligned} \tag{6}$$

3 Conclusion

We will see, that our results will be helpful in future works.

Acknowledgement

Thanks to the private donators who financially support this work.

License

References

[1] O'Donnell, Ryan: *Analysis of boolean functions.* Cambridge University Press, 2014

[2] Wang, Z. ; Gong, G.: Discrete Fourier Transform of Boolean Functions over the Complex Field and Its Applications. In: *IEEE Transactions on Information Theory* 64 (2018), Nr. 4, S. 3000–3009

[3] Zhang, Zhiwei: *Solving Hybrid Boolean Constraints by Fourier Expansions and Continuous Optimization*, Rice University, Diss., 2020

[4] Kyrillidis, Anastasios ; Shrivastava, Anshumali ; Vardi, Moshe Y. ; Zhang, Zhiwei: FourierSAT: A Fourier Expansion-Based Algebraic Framework for Solving Hybrid Boolean Constraints. In: *AAAI*, 2020, S. 1552–1560