SZA :: Signed Zero Algebra

A New Take On Propositional Logic

Thomas W Thorbjørnsen
Eirik Wittersø
Benjamin Benjaminsen
Torstein Nordgård-Hansen

07.04.2019

Contents

Chapter 1

Introduction

1.1 Preface

1.2 Motivation

As one usually does on a Saturday evening, we found ourselves suddenly engrossed in a vivid discussion about the societal value of the number zero. Whilst discussing the differences between null, zero, nil and naught, and their non-evident linguistic difference in Norwegian; a comment was made regarding the existence of both a positive and a negative zero in java. This proposition was by some found preposterous, and prompted an immediate investigation. Upon said investigation, its truthfulness was found evident, further bewildering the doubtful subjects regarding its arithmetical implementation. Addition, subtraction and multiplication was all applied, and the results organized and examined. What thenceforth emerged bore a striking resemblance to Boolean algebra, and the expansion and elaboration of said results is the subject of this paper.

Chapter 2

Finding the Algebraic Structure

2.1 Defining the Object

We will start off by defining all primitive objects. These definitions are a formalisation of how Java interprets the given symbols. We need to define the set which we will perform operations on, and the functions we can apply to this set. We define our set R as the following:

Definition 2.1.1.
R is the set with the elements {0, -0} i.e. $R = \{0, -0\}$

With the set it is a natural question to ask what we can do with this set. Java gave us the following functions which acts on the set. We will define:

Definition 2.1.2.
Let $+, -, \cdot$ be funtions such that $+, -, \cdot : R \times R \to R, \quad (a, b) \mapsto a(\bullet)b$.
These tables are a description of the complete function for each operation:

a	b	a+b
0	0	0
0	-0	0
-0	0	0
-0	-0	-0

a	b	a-b
0	0	0
0	-0	0
-0	0	-0
-0	-0	0

a	b	a · b
0	0	0
0	-0	-0
-0	0	-0
-0	-0	0

By inspecting the functions given by Java, we can see that they are all binary operations. We can also see that the "-" function is composite of $\cdot$ and $+$ (1). It is also possible to define "-" as a unary operation with $\cdot$ (2).

1. $- : R \times R \to R, (a, b) \mapsto a + (-0 \cdot b)$
2. $- : R \to R, a \mapsto -0 \cdot a$

In the rest of this paper, - will be treated as either (1) or (2), if it is not clear from the context what - is it will be stated.

2.2 The One and Only structure, $\mathbb{Z}_2$

2.2.1 Proving the Structure

In the last section we made a set with two functions acting on that set, namely addition and multiplication; and another function, -, defined with the multiplication function. In other words we can say that we have a set and two binary operations, which awfully looks like a ring. Due to the special circumstances of R, there is only one such ring: $\mathbb{Z}_2$.

The idea for this section of this paper is proving that $(r, \cdot, +) \simeq (\mathbb{Z}_2, +_2, \cdot_2)$. After that isomorphism is stated, the rules and behaviors of the ring should be clarified.

Proposition 2.2.1.
Let $R = \{0, -0\}$, and let $+$ and $\cdot$ be the functions defined in the last section, then $(R, \cdot, +)$ is a ring

Lemma 2.2.1.1.
Let S be a set with to binary functions; $+$, $\cdot$. S is a ring if S is isomorphic with a ring R.

I will start of by proving Lemma 2.2.1.1, then use that Lemma on R to show that the proposition holds.

Proof. Let S be a set and $+$, $\cdot$ be two binary operations on S, then let $(R, +_0, \cdot_0)$ be a ring. Assume $S \simeq R$, this means that $\exists$ a function $\phi : S \to R$ which is a ring isomorphism. Since ϕ is a ringisomorphism $\exists \Phi : \phi \circ \Phi = \Phi \circ \phi = id$. Lets start with showing that $(S, +)$ is an abelian group.

1. The identity
   It exists since it exists in R

2. Associativity
   $a + (b + c) = \phi(a' +_0 (b' +_0 c')) = \phi((a' +_0 b') +_0 c') = (a + b) + c$

3. Inverse
   $0 = \phi(0') = \phi(a' + a^{-1\prime}) = a + a^{-1}$

4. Commutativity
   $a + b = \phi(a' + b') = \phi(b' + a') = b' + a'$

Now we need to show that $(S*, \cdot)$ is a monoid. This is done in the exactly same way as addition, so the reader can verify this fact for themselves. □

Proof. In order to prove proposition 2.2.1 we need to find a ringisomorphism $\phi : R \to \mathbb{Z}_2$. This is done by creating a mapping between R and $\mathbb{Z}_2$. We can easily find this mapping by looking at the operation tables.

Tables for R

a	b	a + b
0	0	0
0	-0	0
-0	0	0
-0	-0	-0

a	b	a · b
0	0	0
0	-0	-0
-0	0	-0
-0	-0	0

Tables for $\mathbb{Z}_2$

a	b	a $\cdot_2$ b
0	0	0
0	1	0
1	0	0
1	1	1

a	b	a $+_2$ b
0	0	0
0	1	1
1	0	1
1	1	0

By looking at these tables we observe that table 1 and 3 are identical if -0 gets mapped to 1. The same thing can be said about table 2 and 4. We proceed to define the function $\phi : 0 \mapsto 0, -0 \mapsto 1$. From the tables we observe that addition and multiplication in R respects the axioms for rings, since it is identical to the table for $\mathbb{Z}_2$. □

2.2.2 Identifying the Properties

From this proof of Proposition 2.2.1 we have shown that R is a ring. The most peculiar thing about this ring is that the binary operations don't behave as one would think. Addition in the ring R is equivalent to multiplication in $\mathbb{Z}_2$ and multiplication in R is equivalent to addition in $\mathbb{Z}_2$.

Remark.

1. **multiplication is the additive operation**, and
   addition is the multiplicative operation.
2. $+$ has higher precedence than $\cdot$

Since $+$ and $\cdot$ have switched places in the ring, things might be a bit confusing. Let's state some identites for the ring R. Assume that $a, b, c \in R$ then:

1. $0 + a = 0$
2. $-0 + a = a$
3. $0 \cdot a = a$
4. $-0 \cdot a = -a$
5. $a + a = a$
6. $a \cdot a = 0$
7. $a + (a \cdot b) = a \cdot (a + b) = a - b$
8. $a + (b \cdot c) = (a + b) \cdot (b + c)$
9. $a \cdot (b + c)$ is irreducible

2.2.3 Discussion

Why is it useful discussing the ring R, when it is equivalent to $\mathbb{Z}_2$. Recall how we first defined R. R is defined simply by how Java interprets the different symbols when parsed. The idea behind this is to make a new bridge between computation and represention in $\mathbb{Z}_2$. If we want to calculate in $\mathbb{Z}_2$ on the computer, we need to tell it that this is another algebraic structure we want to use. This system allows us to simply use the given symbols we use on the paper and promptly put it in the computer.

In the rest of this paper the algebra formed by the ring R will be called for **SZA**, short for Signed Zero Algebra.

When it comes to applications of SZA it is possible to draw a connection between SZA and Boolean Algebra. By comparing the truth tables for **addition** and **multiplication** with the truth tables for OR and XNOR a striking resemblance appear. By defining 0 as **true** and -0 as **false** the + turns into OR and $\cdot$ turns into XNOR.

Tables for SZA

a	b	a + b
0	0	0
0	-0	0
-0	0	0
-0	-0	-0

a	b	a $\cdot$ b
0	0	0
0	-0	-0
-0	0	-0
-0	-0	0

Truth tables

a	b	a OR b
T	T	T
T	F	T
F	T	T
F	F	F

a	b	a XNOR b
T	T	T
T	F	F
F	T	F
F	F	T

Chapter 3

Linking SZA to Boolean Algebra

3.1 Building Logic Conjunctions

3.1.1 Primitive Conjunctions

3.1.2 Composite Conjunctions

3.2 How To Do Computations

3.2.1 From SZA to Modular Arithmetic

3.2.2 Proofs of given Composite Conjuctions

Chapter 4

An Application

4.1 Creating a Low Level Arithmetic Machine

4.1.1 Making a Full Bit Adder

4.1.2 Extending the Machine