Boundary Algebra

Eric C.R. Hehner

Department of Computer Science, University of Toronto
hehner@cs.utoronto.ca

A brief introduction to the main idea underlying boundary algebra.

Binary Algebra

We start with a small fragment of binary algebra (also known as boolean algebra). The fragment includes constant $\top$ (true), variables such as x and y, prefix operator $\neg$ (negation), and infix operator $\wedge$ (conjunction). We give $\neg$ higher precedence than $\wedge$, so we need brackets ( ) to negate a conjunction. Here is an example expression:

$$\neg(\neg x\wedge\neg\neg\top)$$

That is all we need; all of binary algebra can be expressed with only these symbols.

Bracket Algebra

Bracket algebra is a simplification of the fragment of binary algebra just presented. To negate an expression, put ( ) around it. For example, (x) is "not x". To conjoin two expressions, just sit them next to each other. For example, what we wrote as

$$\neg(\neg x\wedge\neg\neg y)$$

we now write as

$$((x)((y)))$$

And finally, what we wrote as $\top$ we now write as nothing. For example,

$$\neg(\neg x\wedge\neg\neg\top)$$

becomes

$$((x)(()))$$

Writing $\top$ as nothing raises a question: How do you know where the nothings are, and how many nothings there are? In the expression we just wrote, maybe there's a nothing just before x, and another between)(, and two of them between ( ) . If so, then the bracket algebra expression

$$((x)(()))$$

means the same as the binary algebra expression

$$\neg(\neg(\top\wedge x)\wedge\top\wedge\neg\neg(\top\wedge\top))$$

In binary algebra, $\top$ is the identity for $\wedge$ so $\top\wedge x$ and $x\wedge\top$ are both equivalent to x. In other words, occurrences of $\top\wedge$ and $\wedge\top$ can be added or deleted without changing the meaning of the expression. So the previous expression is equivalent to

$$\neg(\neg x\wedge\neg\neg\top)$$

This is just like adding or deleting occurrences of 0+ and +0 to an arithmetic expression. In bracket algebra, it doesn't matter where or how many nothings there are. Two of them, one of them, even zero of them, all have the same meaning.

The expressions of bracket algebra are all and only those expressions with properly matched pairs of parentheses ( ) , with proper nesting, with variables sprinkled anywhere.

There are three proof rules in bracket algebra. If x , y , and z are any bracket algebra expressions, then

$((x))$	can replace or be replaced by	x	double negation rule
$x(\,)y$	can replace or be replaced by	$(\,)$	base rule
$x\ y\ z$	can replace or be replaced by	$x'\ y\ z'$	context rule

where x' is x with occurrences of y added or deleted, and similarly z' is z with occurrences of y added or deleted. The context rule does not say how many occurrences of y are added or deleted; it could be zero or more. Expressions obtained from each other by following the rules have the same meaning. To prove truth, you just follow the proof rules until the expression disappears. For example, the binary algebra expression

$\neg(\neg a\wedge b\wedge\neg(\neg a\wedge b))$

becomes the bracket algebra expression

$((a)b((a)b))$

Here is a proof of this expression.

	$((a)b((a)b))$	use context rule: empty for x, $(a)b$ for y, $((a)b)$ for z
becomes	$((a)b(\quad))$	use base rule: $(a)b$ for x and empty for y
becomes	$(\quad(\quad))$	use double negation rule
becomes		

Since the last expression is empty, all these expressions express truth. Here is another example. The binary algebra expression

$\neg(\neg(a\wedge b)\wedge\neg(\neg a\wedge b)\wedge\neg(a\wedge\neg b)\wedge\neg(\neg a\wedge\neg b))$

becomes the bracket algebra expression

$((ab)((a)b)(a(b))((a)(b)))$

Here is its proof.

$((ab)((a\ )b)(a(b))((a\quad)(b)))$	context: insert b
$((ab)((ab)b)(a(b))((a\quad)(b)))$	context: delete (ab)
$((ab)(\quad b)(a(b))((a\quad)(b)))$	context: insert (b)
$((ab)(\quad b)(a(b))((a(b))(b)))$	context: delete $(a(b))$
$((ab)(\quad b)(a(b))(\quad (b)))$	context: delete (b) twice
$((ab)(\quad b)(a\quad)(\quad))$	base
$(\quad (\quad))$	double negation
poof	

To prove falsity, follow the rules until the expression becomes $(\,)$.

Interpretations and Abstraction

As presented, we interpret nothing as $\top$ (truth), parentheses as $\neg$ (negation), and adjacency as $\wedge$ (conjunction). There is another, dual way to interpret the symbols and rules of bracket algebra: interpret nothing as $\bot$ (false), parentheses as $\neg$ (negation), and adjacency as $\vee$ (disjunction). In binary algebra, $\bot$ is the identity for $\vee$ so $\bot\vee x$ and $x\vee\bot$ are both equivalent to x . In other words, occurrences of $\bot\vee$ and $\vee\bot$ can be added or deleted without changing the meaning of the expression. In bracket algebra, in either interpretation, it doesn't matter where or how many nothings there are. The rules of proof remain the same in both interpretations; in the dual interpretation, proving falsity means following the rules until the expression disappears, and proving truth means following the proof rules until the expression becomes $(\,)$.

Because the syntax and proof rules remain the same for both interpretations, bracket algebra can be conducted without committing ourselves to either interpretation. It is an abstract algebra. The brackets ( ) just distinguish what is between them from what is outside them. In either interpretation, $x(y)$ is equivalent to $(y)x$; all that's important is that x is outside the brackets, and y is inside the brackets. The brackets make one basic distinction (inside versus outside), and that's all that's needed for binary algebra.

In both interpretations, $a\ b$ is equivalent to $b\ a$. Here is the proof.

$a\ b$	context: nothing for x , a for y , b for z
$a\ b\ a$	context: ab for x , a for y , nothing for z
$b\ a$	

Box Algebra

Bracket algebra, like binary algebra, and most of mathematics, is one-dimensional. Expressions are formed as a sequence of symbols. We now move up to two dimensions. A plane box distinguishes what's inside it from what's outside it. That's enough for an algebra. The boxes can be anywhere in the plane, as long as they are properly nested, one inside the other, or properly beside each other, not overlapping. In other words, box boundaries cannot cross each other. Variables can be anywhere. And you can say there are as many nothings as you want anywhere you want. In one interpretation, the nothings represent truth; putting things (nothings, variables, and boxes) in the same space (you can draw a line from one thing to the other without crossing a box boundary) means conjunction; and a box negates what is inside it. In the other interpretation, the nothings represent falsity; putting things in the same space means disjunction; and again a box negates what is inside it. You can move anything around anywhere in its space (not crossing a box boundary) without changing the meaning. The proof rules need to be reworded to apply to two dimensions. Proofs can be conducted without choosing an interpretation.

Boundary Algebra

Obviously, it doesn't matter if the shape of the boundary is rectangular, circular, or irregular; what matters is that it distinguishes what's inside it from what's outside it. And we could move to higher-dimensional spaces and tell the same story. So that's boundary algebra. But there is so much more to say.

Boundary algebra is the work of Philip Meguire of the University of Canterbury in New Zealand. He has written a 150-page book titled "*Boundary Algebra: a Simpler Approach to Boolean Algebra and Sentential Logic*", 2019. He owes some of his inspiration to George Spencer-Brown: *Laws of Form*, 1969. Before that, Charles Sanders Peirce (1839-1914) had the basic idea. The bracket notation is due to C.Croskin: *Ways of Knowing*, 1978.

I feel obliged to mention the Plex philosophy of Douglas T. Ross (1929-2007). Doug tried to explain it to me many times, but I never understood any of it. Doug quoted Spencer-Brown and Peirce, so I wonder if Plex might be related to boundary algebra.