

# Using *Mathematica* to Understand the Computer Proof of the Robbins Conjecture

Branden Fitelson

`fitelson@students.wisc.edu`

Department of Philosophy  
University of Wisconsin–Madison  
600 North Park Street  
Madison, WI 53706

December 14, 1997

## ■ Introduction<sup>1</sup>

In October 1996, Bill McCune *et al* at Argonne National Laboratory made national headlines (see Kolata (1996) and Peterson (1997)) with their computer proof of the long-standing Robbins conjecture. The Robbins conjecture had stumped some of the world's best mathematicians for over 60 years. Amazingly, the Argonne team's automated theorem-proving program **EQP** took only 8 days to find a proof of it. Unfortunately, the proof found by **EQP** is quite complex and difficult to follow. Some of the steps of the **EQP** proof require highly complex and unintuitive substitution strategies. As a result, it is nearly impossible to reconstruct or verify the computer proof of the Robbins conjecture entirely by hand. This is where the unique symbolic capabilities of *Mathematica* 3 come in handy. With the help of *Mathematica*, it is relatively easy to work out and explain each step of the dense **EQP** proof in detail. In this paper, I use *Mathematica* to provide a detailed, step-by-step reconstruction of the highly complex **EQP** proof of the Robbins conjecture.

## ■ Preliminaries: Boolean Algebra—Definition and Notation

The first strictly axiomatic definitions of Boolean algebra were surveyed by E.V. Huntington (1904).<sup>2</sup> Huntington (1904) presents four axiomatizations of Boolean algebra. The first Huntington axiomatization, which is closest to Boole's original system (Kneale and Kneale 1962, page 423), and which is often taken as definitive of Boolean algebra (Stoll 1963, page 249), is as follows

**Definition.** A *Boolean algebra* is a 6-tuple  $\langle B, \oplus, \otimes, \bar{\cdot}, \mathbf{0}, \mathbf{1} \rangle$ , where  $B$  is a set,  $\oplus$  is a binary relation (usually called *union*, *join*, or *disjunction*) in  $B$ ,  $\otimes$  is a binary relation in  $B$  (usually called *intersection*, *meet*, or *conjunction*),  $\bar{\cdot}$  is a function from  $B$  into  $B$  (usually called *complementation* or *negation*),  $\mathbf{0}$  and  $\mathbf{1}$  are distinct elements of  $B$ , and the following five conditions are satisfied

(i) The operations  $\oplus$  and  $\otimes$  are *associative*: for all  $a, b, c \in B$

$$a \oplus (b \oplus c) = (a \oplus b) \oplus c \quad \text{and} \quad a \otimes (b \otimes c) = (a \otimes b) \otimes c$$

(ii) The operations  $\oplus$  and  $\otimes$  are *commutative*: for all  $a, b \in B$

$$a \oplus b = b \oplus a \quad \text{and} \quad a \otimes b = b \otimes a$$

(iii) Each of the operations  $\oplus$  and  $\otimes$  *distributes* over the other: for all  $a, b, c \in B$

$$a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c) \quad \text{and} \quad a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$$

(iv) For all  $a \in B$ ,

$$a \oplus \mathbf{0} = a \quad \text{and} \quad a \otimes \mathbf{1} = a$$

(v) For each  $a \in B$ , there exists an element  $\bar{a} \in B$  such that

$$a \oplus \bar{a} = \mathbf{1} \quad \text{and} \quad a \otimes \bar{a} = \mathbf{0}$$

In the context of Boolean *logic*, for instance,  $\oplus$  is *disjunction* (typically denoted by  $\vee$ ),  $\otimes$  is *conjunction* (typically denoted by  $\wedge$ ), and  $\bar{\cdot}$  is *negation* (typically denoted by  $\neg$ ). Moreover, in Boolean logic, the elements  $\mathbf{0}$  and  $\mathbf{1}$  denote *contradiction* (e.g.,  $p \wedge \neg p$ ) and *tautology* (e.g.,  $p \vee \neg p$ ), respectively. With these terminological preliminaries out of the way, we are ready to begin our brief sketch of the history of the Robbins conjecture.

## ■ A Brief History of the Robbins Conjecture

Almost thirty years after his seminal 1904 paper, Huntington (1933a and 1933b) discovered the following greatly simplified three axiom basis for Boolean algebra, which only requires the *single* binary relation  $\oplus$ .<sup>3</sup>

$$\begin{array}{ll} x \oplus y = y \oplus x & \text{[Commutativity of } \oplus \text{]} \\ (x \oplus y) \oplus z = x \oplus (y \oplus z) & \text{[Associativity of } \oplus \text{]} \\ \overline{\overline{x \oplus y \oplus \overline{x \oplus y}}} = x & \text{[Huntington equation]} \end{array}$$

Shortly thereafter, Herbert Robbins conjectured that if the Huntington equation were replaced with the following equation, which, in any Boolean algebra, is *equivalent* to the Huntington equation (see the *Appendix* for a one-line *Mathematica* proof of this equivalence)

$$\overline{\overline{\overline{x \oplus y \oplus x \oplus y}}} = x \quad \text{[Robbins equation]}$$

then the resulting axiomatization would *also* be characteristic of Boolean algebra. This is the *Robbins conjecture*. Robbins and Huntington could not find a proof of (or a counterexample to) the Robbins conjecture. The problem was later studied intensively by Tarski and his students (Tarski *et al* 1971, problem 1.1).

Algebras satisfying commutativity, associativity, and the Robbins equation are known as *Robbins algebras*. It is easy to show that every Boolean algebra is a Robbins algebra (see the *Appendix* for an elegant *Mathematica* proof of this), so the interesting question is "Are all Robbins algebras Boolean?". In other words, can the Huntington equation be derived from commutativity, associativity, and the Robbins equation?

Until the late 1970's, very little progress was made toward resolving the Robbins conjecture. In 1979, S. Winker at Argonne National Laboratory began looking for Boolean conditions that are sufficient to make any Robbins algebra Boolean. With the help of Argonne's automated theorem-provers, many such sufficient Boolean conditions were found. For instance, the *double negation* condition

$$\overline{\overline{x}} = x \quad \text{[Double negation]}$$

can easily be shown to make any Robbins algebra Boolean (see the *Appendix* for a *Mathematica* proof). Unfortunately, these first Boolean conditions found by Winker proved to be *too strong*, in the sense that they were almost as difficult to deduce from the Robbins axioms as the Huntington equation itself! As a result, these early Winker conditions (*e.g.*, double negation) were not very useful for resolving the Robbins conjecture. So, Winker set out to find logically weaker sufficient conditions. Ultimately, Winker (1992) was able to show that the surprisingly weak condition

$$(\exists C)(\exists D) \overline{\overline{C \oplus D}} = \overline{C} \quad \text{[Winker condition]}$$

also suffices, with proofs by hand using insight from theorem-prover searches.

Since then, the problem has received continued attention from mathematicians and computer scientists, but until October 1996, no significant mathematical progress on the problem was made.

## ■ The Computer Proof of the Robbins Conjecture

The proof that solves the Robbins problem was found on October 10, 1996, by Bill McCune *et al* at Argonne, using their mechanical theorem-prover **EQP**.<sup>4</sup> **EQP** is similar in many ways to the more well known Argonne theorem-prover **Otter**.<sup>5</sup> **EQP** proved that every Robbins algebra satisfies the Winker condition, by refuting the following pair of assumptions

$$\begin{array}{ll} \overline{\overline{x \oplus y \oplus x \oplus y}} = x & \text{[Robbins equation]} \\ \overline{x \oplus y} \neq \overline{\overline{x}} & \text{[Denial of the Winker condition]} \end{array}$$

It follows immediately from this and Winker's work that *all Robbins algebras are Boolean (i.e., that the Robbins conjecture is true)*. The successful proof search took about 8 days on an RS/6000 computer and used about 30 megabytes of memory.

Unfortunately, the 16-line proof object generated by **EQP** is very cryptic and complex. Moreover, **EQP** does not provide any information about *how* each line in the proof object follows from previous lines. This makes verifying the **EQP** proof object quite a daunting task for a mere human. Below, I will illustrate how *Mathematica* can be used to make human verification and explanation of the **EQP** proof object more tractable.

## ■ The EQP Proof Object: EQP Notation vs *Mathematica* 3 Notation

The main problem with the **EQP** proof object is that it uses the one-dimensional notation "*n(x)*" instead of Boole's two-dimensional notation " $\overline{x}$ " for denoting the complementation (or negation) operator.<sup>6</sup> This makes it almost impossible to keep track of the highly complex and deeply nested syntactical patterns that occur in the computer proof of the Robbins conjecture. By allowing us to work directly with Boole's more visually perspicuous 2-D notation, *Mathematica* 3 makes it much easier for us to see what is going on in the **EQP** proof.

Below, I have included the entire 16-line **EQP** proof object, first in **EQP** notation, then in *Mathematica* 3 notation. Notice how difficult it is to comprehend (or even parse!) the lines written in **EQP** notation, as compared to *Mathematica* 3 notation (especially, the extremely complex line 15). And, notice how little **EQP** tells us about *how* each line can be derived from previous lines. The bracketed lists of line numbers after each line tell us which previous lines in the proof were used in the derivation of the present line. **EQP** tells us *nothing* about *how* these derivations were performed (*i.e.*, *which* substitutions were made in which lines, *etc.*). These "hints" are very sketchy. But, as I will demonstrate below, these sketchy "hints" provide sufficient information for us (with a lot of help from *Mathematica*) to reconstruct a detailed proof.

## ■ The EQP Proof Object in EQP Notation<sup>7</sup>

1.  $n(n(x \oplus y) \oplus n(y \oplus n(x))) == y$  [Assumption: the Robbins equation]
2.  $n(y \oplus n(y \oplus n(x) \oplus n(x \oplus y))) == n(x \oplus y)$  [1]
3.  $n(y \oplus n(x \oplus y \oplus n(y \oplus n(x)))) == n(y \oplus n(x))$  [1]
4.  $n(n(y \oplus n(x)) \oplus n(x \oplus y \oplus y \oplus n(y \oplus n(x)))) == y$  [1, 3]
5.  $n(n(y \oplus z) \oplus n(z \oplus n(y \oplus n(x)) \oplus n(x \oplus y \oplus y \oplus n(y \oplus n(x)))))) == z$  [1, 4]
6.  $n(y \oplus n(x \oplus y \oplus y \oplus n(y \oplus n(x)) \oplus n(y \oplus n(x)))) == n(y \oplus n(x))$  [1, 4]
7.  $n(n(y \oplus n(x)) \oplus n(x \oplus y \oplus y \oplus y \oplus n(y \oplus n(x)) \oplus n(y \oplus n(x)))) == y$  [1, 6]
8.  $n(z \oplus n(z \oplus n(y \oplus z) \oplus n(y \oplus n(x)) \oplus n(x \oplus y \oplus y \oplus n(y \oplus n(x)))))) == n(y \oplus z)$  [1, 5]
9.  $n(n(u \oplus n(y \oplus z)) \oplus n(u \oplus z \oplus n(z \oplus n(y \oplus z) \oplus n(y \oplus n(x)) \oplus n(x \oplus y \oplus y \oplus n(y \oplus n(x)))))) == u$  [1, 8]
10.  $n(x \oplus n(x \oplus x \oplus x \oplus n(x \oplus n(x)))) == n(x \oplus n(x))$  [1, 2, 9]
11.  $n(n(y \oplus n(x \oplus n(x))) \oplus n(x \oplus y \oplus n(x \oplus x \oplus x \oplus n(x \oplus n(x)))))) == y$  [1, 10]
12.  $n(n(x \oplus n(x)) \oplus n(x \oplus x \oplus x \oplus x \oplus n(x \oplus n(x)))) == x$  [1, 10]
13.  $n(x \oplus n(x \oplus x \oplus x \oplus x \oplus n(x \oplus n(x)) \oplus n(x \oplus n(x)))) == n(x \oplus n(x))$  [1, 12]
14.  $n(n(y \oplus n(x \oplus n(x))) \oplus n(x \oplus y \oplus n(x \oplus x \oplus x \oplus x \oplus n(x \oplus n(x)) \oplus n(x \oplus n(x)))))) == y$  [1, 13]
15.  $n(x \oplus n(x \oplus n(x \oplus x \oplus x \oplus n(x \oplus n(x))) \oplus n(x \oplus x \oplus x \oplus x \oplus n(x \oplus n(x)) \oplus n(x \oplus n(x)))))) ==$   
 $n(x \oplus x \oplus x \oplus x \oplus n(x \oplus n(x)) \oplus n(x \oplus n(x)))$  [7, 11]
16.  $n(x \oplus x \oplus x \oplus x \oplus n(x \oplus n(x)) \oplus n(x \oplus n(x))) == n(x \oplus x \oplus x \oplus n(x \oplus n(x)))$  [4, 14, 15]

### ■ The EQP Proof Object in *Mathematica* 3 Notation

$$1. \overline{\overline{\overline{x \oplus y \oplus y \oplus \overline{x}}}} = y \quad [\text{Assumption: the Robbins Equation}]$$

$$2. \overline{\overline{\overline{y \oplus y \oplus \overline{x \oplus x \oplus y}}}} = \overline{x \oplus y} \quad [1]$$

$$3. \overline{\overline{\overline{y \oplus x \oplus y \oplus y \oplus \overline{x}}}} = \overline{y \oplus \overline{x}} \quad [1]$$

$$4. \overline{\overline{\overline{\overline{y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}} = y \quad [1, 3]$$

$$5. \overline{\overline{\overline{\overline{\overline{y \oplus z \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}}}} = z \quad [1, 4]$$

$$6. \overline{\overline{\overline{\overline{y \oplus x \oplus y \oplus y \oplus y \oplus \overline{x} \oplus y \oplus \overline{x}}}}} = \overline{y \oplus \overline{x}} \quad [1, 4]$$

$$7. \overline{\overline{\overline{\overline{\overline{y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus y \oplus \overline{x} \oplus y \oplus \overline{x}}}}}}} = y \quad [1, 6]$$

$$8. \overline{\overline{\overline{\overline{\overline{z \oplus z \oplus y \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}}}} = \overline{y \oplus z} \quad [1, 5]$$

$$9. \overline{\overline{\overline{\overline{\overline{\overline{u \oplus y \oplus z \oplus u \oplus z \oplus z \oplus y \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}}}}}} = u \quad [1, 8]$$

$$10. \overline{\overline{\overline{\overline{x \oplus x \oplus x \oplus x \oplus x \oplus \overline{x}}}}} = \overline{x \oplus \overline{x}} \quad [1, 2, 9]$$

$$11. \overline{\overline{\overline{\overline{\overline{y \oplus x \oplus \overline{x} \oplus x \oplus y \oplus x \oplus x \oplus x \oplus x \oplus \overline{x}}}}}}} = y \quad [1, 10]$$

$$12. \overline{\overline{\overline{\overline{x \oplus \overline{x} \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus \overline{x}}}}} = x \quad [1, 10]$$

$$13. \overline{\overline{\overline{\overline{x \oplus x \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus x \oplus \overline{x}}}}} = \overline{x \oplus \overline{x}} \quad [1, 12]$$

$$14. \overline{\overline{\overline{\overline{\overline{y \oplus x \oplus \overline{x} \oplus x \oplus y \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus x \oplus \overline{x}}}}}}} = y \quad [1, 13]$$

$$15. \overline{\overline{\overline{\overline{\overline{\overline{x \oplus x \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus x \oplus \overline{x}}}}}}}}} = \overline{\overline{\overline{\overline{x \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus x \oplus \overline{x}}}}} \quad [7, 11]$$

$$16. \overline{\overline{\overline{\overline{\overline{x \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} \oplus x \oplus \overline{x}}}}} = \overline{x \oplus x \oplus x \oplus x \oplus \overline{x}} \quad [4, 14, 15]$$

## ■ Using *Mathematica* to Decipher the EQP Proof Object

### ■ Encoding The Robbins Axioms in *Mathematica* 3

#### ■ Encoding the Commutativity and Associativity of $\oplus$

We can tell *Mathematica* to assume *only* that  $\oplus$  is commutative (*viz.*, **Orderless**) and associative (*viz.*, **Flat**), as follows.<sup>8</sup>

```
In[1]:= Attributes[CirclePlus] = {Orderless, Flat};
```

In my reconstruction of the EQP proof, I will just let *Mathematica* automatically make use of the commutativity and associativity of  $\oplus$ . I will not explicitly announce which steps depend on these two basic properties of  $\oplus$ . This will allow us to focus our attention on the more interesting and important logical role played by the Robbins equation itself.<sup>9</sup>

#### ■ Encoding the Robbins Equation—Two Handy *Mathematica* Constructs

The two-place function **RobbinsExpand** allows us to logically expand  $x$  (in terms of *any*  $y$ ) in accordance with the Robbins equation. In other words, for any  $x$  and  $y$ ,  $\text{RobbinsExpand}[x, y] = \overline{\overline{x \oplus y} \oplus \overline{x \oplus y}} = x$ . Thus, **RobbinsExpand** $[x, y]$  allows us to create arbitrary Robbins-equivalents of any formula  $x$ . The one-place function **RobbinsSimplify** automatically simplifies *Mathematica* expressions containing instances of the Robbins pattern.

```
In[2]:= RobbinsExpand[x_, y_] =  $\overline{\overline{x \oplus y} \oplus \overline{x \oplus y}}$ ;
RobbinsSimplify[expr_] := expr //.  $\overline{\overline{x \oplus y} \oplus \overline{x \oplus y}}$  -> x;
```

Here is a simple example that illustrates how **RobbinsExpand** and **RobbinsSimplify** allow us to expand/simplify *Mathematica* expressions in accordance with the Robbins equation.

```
In[3]:= RobbinsExpand[a, b]  $\oplus$  RobbinsExpand[b, a]
```

```
Out[3]=  $\overline{\overline{a \oplus b \oplus a \oplus b} \oplus \overline{a \oplus b \oplus b \oplus a}}$ 
```

```
In[4]:= % // RobbinsSimplify
```

```
Out[4]= a  $\oplus$  b
```

Now that we have encoded commutativity, associativity and the Robbins equation in *Mathematica* 3, we are ready to tackle the EQP proof of the Winker Condition, line-by-line.

## ■ Deciphering the EQP Proof of the Winker Condition Line–By–Line

### ■ Line 1 of the EQP Proof—The Robbins Equation

Line 1 of the EQP proof is just the Robbins equation, as discussed above

```
In[5]:= ProofLine[1] =  $\overline{\overline{\overline{x \oplus \overline{y \oplus x}}}}$  == y
```

```
Out[5]=  $\overline{\overline{\overline{x \oplus \overline{y \oplus x}}}}$  == y
```

### ■ Line 2 of the EQP Proof—The first (of many) cryptic substitution strategies

Line 2 of the EQP proof is

```
In[6]:= ProofLine[2] =  $\overline{\overline{\overline{y \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}$  ==  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ 
```

```
Out[6]=  $\overline{\overline{\overline{y \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}$  ==  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ 
```

Line 2 of the proof says that  $\overline{\overline{\overline{x \oplus \overline{y}}}} = \overline{\overline{\overline{y \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}$ . All we are told by EQP is that line 2 follows (*somehow*) from the Robbins equation (*i.e.*, line 1). The only way to go from the Robbins equation to line 2 is by creating a Robbins-equivalent of  $\overline{\overline{\overline{x \oplus \overline{y}}}}$  using **RobbinsExpand** [ $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ,  $\mathbf{y}$ ], for *some*  $y$ . The question is *which*  $y$  is such that **RobbinsExpand** [ $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ,  $\mathbf{y}$ ] yields the desired formula  $\overline{\overline{\overline{y \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}$ ?

As with any good proof-theoretic question of this kind, the answer involves a neat trick. In this case, the relevant trick is very difficult to figure out (or even *see*) without *Mathematica's* help. *Mathematica* enabled me to experiment with many candidate formulas in a very short period of time (a task that would be nearly impossible to undertake with pencil and paper). After trying several substitution strategies, I discovered the following crucial Robbins-equivalent of  $\overline{\overline{\overline{x \oplus \overline{y}}}}$

```
In[7]:= RobbinsExpand [ $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ,  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ]
```

```
Out[7]=  $\overline{\overline{\overline{\overline{\overline{\overline{x \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}}}}}$ 
```

The two-dimensional *Mathematica* notation makes it relatively easy to see that the expression **RobbinsExpand** [ $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ,  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ] contains *two* distinct instances of the Robbins pattern. By definition, the *entire* expression is an instance of the Robbins pattern (equivalent to  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ). But, so is the *sub-expression*  $\overline{\overline{\overline{x \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}}$ , which is equivalent to  $y$ . Using **RobbinsSimplify**, we can *automatically* simplify this sub-expression, yielding the desired result.

```
In[8]:= RobbinsExpand [ $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ,  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ] // RobbinsSimplify
```

```
Out[8]=  $\overline{\overline{\overline{y \oplus \overline{y \oplus \overline{x \oplus \overline{x \oplus \overline{y}}}}}}$ 
```

In other words, the left hand side of line 2 can be obtained by logically expanding  $\overline{\overline{\overline{x \oplus \overline{y}}}}$  in terms of  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ , in accordance with the Robbins equation, and then simplifying the resulting expression which contains an instance of the Robbins pattern. All of this can be done (and verified) *automatically* with the following single line of *Mathematica* code.

```
In[9]:= ProofLine[2] == (RobbinsExpand [ $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ,  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ] ==  $\overline{\overline{\overline{x \oplus \overline{y}}}}$ ) // RobbinsSimplify
```

```
Out[9]= True
```

### ■ Line 3 of the EQP Proof

Line 3 of the EQP proof is



```
In[10]:= ProofLine[3] =  $\overline{y \oplus x \oplus y \oplus y \oplus \bar{x}} == \overline{y \oplus \bar{x}}$ 
```

```
Out[10]=  $\overline{y \oplus x \oplus y \oplus y \oplus \bar{x}} == \overline{y \oplus \bar{x}}$ 
```

Seeing how line 3 follows from line 1 (*i.e.*, the Robbins equation) involves a trick very similar to the trick described above in connection with our deciphering of line 2. Here, the trick is to logically expand  $\overline{y \oplus \bar{x}}$  in terms of  $y \oplus x$ , in accordance with the Robbins equation, and then simplify the resulting expression which contains an instance of the Robbins pattern. Once again, this can be done (and verified) *automatically* with the following single line of *Mathematica* code.

```
In[11]:= ProofLine[3] == (RobbinsExpand[ $\overline{y \oplus \bar{x}}$ ,  $y \oplus x$ ] ==  $\overline{y \oplus \bar{x}}$ ) // RobbinsSimplify
```

```
Out[11]= True
```

Equivalently, we can use **RobbinsSimplify** alone to see that line 3 is the following substitution instance of line 1

```
In[12]:= ProofLine[3] == (ProofLine[1] /. { $x \rightarrow y \oplus x$ ,  $y \rightarrow \overline{y \oplus \bar{x}}$ }) // RobbinsSimplify
```

```
Out[12]= True
```

The function **Line3Expand** allows us to rewrite expressions of the form  $\overline{y \oplus \bar{x}}$  in the more complex, line 3-equivalent form  $\overline{y \oplus x \oplus y \oplus y \oplus \bar{x}}$ . **Line3Expand** will come in handy in our proof of line 4, below.

```
In[13]:= Line3Expand[ $\overline{y \oplus \bar{x}}$ ] :=  $\overline{y \oplus x \oplus y \oplus y \oplus \bar{x}}$ ;
```

## ■ Line 4 of the EQP Proof

Line 4 of the EQP proof is

```
In[14]:= ProofLine[4] =  $\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x}} == y$ 
```

```
Out[14]=  $\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x}} == y$ 
```

The EQP proof object says only that line 4 follows from from lines 1 and 3. With our functions **RobbinsExpand**, **RobbinsSimplify**, and **Line3Expand**, we can automatically verify that line 4 is equivalent to

```
In[15]:= ProofLine[4] == (Line3Expand[RobbinsExpand[y, x]] == y) // RobbinsSimplify
```

```
Out[15]= True
```

The trick here is to recognize that the left-hand side of line 1 (*i.e.*,  $\overline{x \oplus \bar{y} \oplus y \oplus \bar{x}}$ ) is *itself* an instance of the right-hand side of line 3

(*i.e.*,  $\overline{y \oplus \bar{x}}$ , where  $y = \overline{y \oplus \bar{x}}$ , and  $x = x \oplus y$ ). This allows us to rewrite  $\overline{x \oplus \bar{y} \oplus y \oplus \bar{x}} = y$  equivalently as

$\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus \bar{x}} = y$ , in accordance with line 3. Finally, the Robbins equation allows us to simplify

$\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus \bar{x}} = y$  down to the desired identity  $\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus \bar{x}} = y$  (we did this *automatically* using **RobbinsSimplify**).

### ■ Line 5 of the EQP Proof

Line 5 of the EQP proof is

```
In[16]:= ProofLine[5] =  $\overline{\overline{\overline{y \oplus z \oplus z \oplus y \oplus \overline{x \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}}$  == z
```

```
Out[16]=  $\overline{\overline{\overline{y \oplus z \oplus z \oplus y \oplus \overline{x \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}}$  == z
```

The EQP proof object says only that line 5 can be obtained via *some* substitution strategy involving lines 1 and 4. To see the appropriate substitutions, let  $a = \overline{y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}$ . Then, **ProofLine[5]** becomes

```
In[17]:= ProofLine[5]' = ProofLine[5] //. { $\overline{\overline{\overline{y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}$  → a}
```

```
Out[17]=  $\overline{\overline{\overline{y \oplus z \oplus z \oplus a}}}$  == z
```

From line 4, we know that  $y = \overline{y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}} = \overline{a}$ . Substituting  $\overline{a}$  for  $y$  in **ProofLine[5]'** yields

```
In[18]:= ProofLine[5]'' = ProofLine[5]' //. {y → a}
```

```
Out[18]=  $\overline{\overline{\overline{z \oplus a \oplus z \oplus a}}}$  == z
```

And, **ProofLine[5]''** is an obvious instance of line 1 (*i.e.*, the Robbins equation).

### ■ Line 6 of the EQP Proof

Line 6 of the EQP proof is

```
In[19]:= ProofLine[6] =  $\overline{\overline{\overline{y \oplus x \oplus y \oplus y \oplus y \oplus \overline{x \oplus y \oplus \overline{x}}}}}$  ==  $\overline{y \oplus \overline{x}}$ 
```

```
Out[19]=  $\overline{\overline{\overline{y \oplus x \oplus y \oplus y \oplus y \oplus \overline{x \oplus y \oplus \overline{x}}}}}$  ==  $\overline{y \oplus \overline{x}}$ 
```

The EQP proof object says only that line 6 can be obtained via *some* substitution strategy involving lines 1 and 4. To see the relevant substitutions, let  $b = x \oplus y \oplus y \oplus \overline{y \oplus \overline{x}}$ . Then, **ProofLine[6]** becomes

```
In[20]:= ProofLine[6]' = ProofLine[6] //. { $\overline{\overline{\overline{x \oplus y \oplus y \oplus y \oplus \overline{x}}}}$  → b}
```

```
Out[20]=  $\overline{\overline{\overline{y \oplus b \oplus y \oplus \overline{x}}}}$  ==  $\overline{y \oplus \overline{x}}$ 
```

From line 4, we know that  $y = \overline{y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}} = \overline{b \oplus y \oplus \overline{x}}$ . Substituting  $\overline{b \oplus y \oplus \overline{x}}$  for the first instance of  $y$  in **ProofLine[6]'** yields

```
In[21]:= ProofLine[6]'' =  $\overline{\overline{\overline{\overline{b \oplus y \oplus \overline{x} \oplus b \oplus y \oplus \overline{x}}}}}$  ==  $\overline{y \oplus \overline{x}}$ 
```

```
Out[21]=  $\overline{\overline{\overline{b \oplus y \oplus \overline{x} \oplus b \oplus y \oplus \overline{x}}}}$  ==  $\overline{y \oplus \overline{x}}$ 
```

And, `ProofLine[6]''` is a substitution instance of line 1, as the test below confirms.

```
In[22]:= ProofLine[6]'' // RobbinsSimplify
Out[22]= True
```

### ■ Line 7 of the EQP Proof

Line 7 of the EQP proof is

```
In[23]:= ProofLine[7] =  $\overline{\overline{\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x} \oplus y \oplus \bar{x}}}}$  == y
Out[23]=  $\overline{\overline{\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x} \oplus y \oplus \bar{x}}}}$  == y
```

The EQP proof object says only that line 7 follows from lines 1 and 6. To see how, let  $c = x \oplus y \oplus y \oplus \overline{\overline{y \oplus \bar{x} \oplus y \oplus \bar{x}}}$ . Then, line 7 becomes

```
In[24]:= ProofLine[7]' = ProofLine[7] //. { $x \oplus y \oplus y \oplus \overline{\overline{y \oplus \bar{x} \oplus y \oplus \bar{x}}} \rightarrow c$ }
Out[24]=  $\overline{\overline{\overline{y \oplus c \oplus y \oplus \bar{x}}}}$  == y
```

From line 6, we know that  $\overline{\overline{y \oplus \bar{x}}} = \overline{\overline{y \oplus x \oplus y \oplus y \oplus \overline{\overline{y \oplus \bar{x} \oplus y \oplus \bar{x}}}}} = \overline{\overline{y \oplus c}}$ . Substituting  $\overline{\overline{y \oplus c}}$  for  $\overline{\overline{y \oplus \bar{x}}}$  in `ProofLine[7]'` yields

```
In[25]:= ProofLine[7]'' = ProofLine[7]' //. { $\overline{\overline{y \oplus \bar{x}}} \rightarrow \overline{\overline{y \oplus c}}$ }
Out[25]=  $\overline{\overline{\overline{y \oplus c \oplus y \oplus c}}}$  == y
```

which is an obvious instance of the Robbins equation.

### ■ Line 8 of the EQP Proof

Line 8 of the EQP proof is

```
In[26]:= ProofLine[8] =  $\overline{\overline{\overline{z \oplus z \oplus \overline{\overline{y \oplus \bar{z} \oplus y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x}}}}}}$  ==  $\overline{\overline{y \oplus \bar{z}}}$ 
Out[26]=  $\overline{\overline{\overline{z \oplus z \oplus \overline{\overline{y \oplus \bar{z} \oplus y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x}}}}}}$  ==  $\overline{\overline{y \oplus \bar{z}}}$ 
```

The EQP proof says only that line 8 can be obtained via *some* substitution strategy involving lines 1 and 5. To see the relevant strategy, let  $d = \overline{\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x}}}$ . Then, `ProofLine[8]` becomes

```
In[27]:= ProofLine[8]' = ProofLine[8] //. { $\overline{\overline{y \oplus \bar{x} \oplus x \oplus y \oplus y \oplus y \oplus \bar{x}}} \rightarrow d$ }
Out[27]=  $\overline{\overline{\overline{z \oplus z \oplus d \oplus \overline{\overline{y \oplus \bar{z}}}}}}$  ==  $\overline{\overline{y \oplus \bar{z}}}$ 
```

From line 5, we know that  $z = \overline{\overline{\overline{y \oplus z \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}} = \overline{\overline{\overline{y \oplus z \oplus z \oplus \overline{d}}}}$ . Substituting  $\overline{\overline{\overline{y \oplus z \oplus z \oplus \overline{d}}}}$  for the first instance of  $z$  in **ProofLine[8]'** yields

```
In[28]:= ProofLine[8]'' =  $\overline{\overline{\overline{y \oplus z \oplus z \oplus \overline{d} \oplus z \oplus \overline{d} \oplus y \oplus z}}} == \overline{\overline{\overline{y \oplus z}}}$ 
```

```
Out[28]=  $\overline{\overline{\overline{y \oplus z \oplus z \oplus \overline{d} \oplus z \oplus \overline{d} \oplus y \oplus z}}} == \overline{\overline{\overline{y \oplus z}}}$ 
```

**ProofLine[8]''** is the following substitution instance of line 1.

```
In[29]:= ProofLine[8]'' == (ProofLine[1] /. {y ->  $\overline{\overline{\overline{y \oplus z}}}$ , x ->  $z \oplus \overline{d}$ })
```

```
Out[29]= True
```

### ■ Line 9 of the EQP Proof

Line 9 of the EQP proof is

```
In[30]:= ProofLine[9] =  $\overline{\overline{\overline{u \oplus \overline{y \oplus z} \oplus u \oplus z \oplus z \oplus \overline{y \oplus z} \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}} == u$ 
```

```
Out[30]=  $\overline{\overline{\overline{u \oplus \overline{y \oplus z} \oplus u \oplus z \oplus z \oplus \overline{y \oplus z} \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}} == u$ 
```

The EQP proof says only that line 9 follows somehow from lines 1 and 8. To see how, let  $e = \overline{\overline{\overline{y \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}$ . Then, **ProofLine[9]** becomes

```
In[31]:= ProofLine[9]' = ProofLine[9] //. { $\overline{\overline{\overline{y \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}}$  -> e}
```

```
Out[31]=  $\overline{\overline{\overline{u \oplus \overline{y \oplus z} \oplus u \oplus z \oplus z \oplus e}}} == u$ 
```

From line 8, we know that  $\overline{\overline{\overline{y \oplus z}}} = z \oplus z \oplus \overline{\overline{\overline{y \oplus z \oplus y \oplus \overline{x} \oplus x \oplus y \oplus y \oplus y \oplus \overline{x}}}} = z \oplus z \oplus e$ . Substituting  $z \oplus z \oplus e$  for  $\overline{\overline{\overline{y \oplus z}}}$  in **ProofLine[9]'** yields

```
In[32]:= ProofLine[9]'' = ProofLine[9]' //. { $\overline{\overline{\overline{y \oplus z}}} \rightarrow z \oplus z \oplus e$ }
```

```
Out[32]=  $\overline{\overline{\overline{u \oplus z \oplus z \oplus e \oplus u \oplus z \oplus z \oplus e}}} == u$ 
```

**ProofLine[9]''** is the following substitution instance of line 1.

```
In[33]:= ProofLine[9]'' == (ProofLine[1] /. {y -> u, x ->  $z \oplus z \oplus e$ })
```

```
Out[33]= True
```

### ■ Line 10 of the EQP Proof

Line 10 of the EQP proof is



```
In[41]:= ProofLine[10]'' == ProofLine[9]***
Out[41]= True
```

### ■ Line 11 of the EQP Proof

Line 11 of the EQP proof is

```
In[42]:= ProofLine[11] =  $\overline{y \oplus x \oplus \bar{x} \oplus x \oplus y \oplus x \oplus x \oplus x \oplus x \oplus \bar{x}}$  == y
Out[42]=  $\overline{y \oplus x \oplus \bar{x} \oplus x \oplus y \oplus x \oplus x \oplus x \oplus x \oplus \bar{x}}$  == y
```

The EQP proof says only that line 11 can be derived from lines 1 and 10. To see how, let  $g = \overline{x \oplus x \oplus x \oplus x \oplus \bar{x}}$ . Then, **ProofLine[11]** becomes

```
In[43]:= ProofLine[11]' = ProofLine[11] /. { $\overline{x \oplus x \oplus x \oplus x \oplus \bar{x}} \rightarrow g$ }
Out[43]=  $\overline{y \oplus x \oplus \bar{x} \oplus \bar{x} \oplus y \oplus g}$  == y
```

From line 10, we know that  $\overline{x \oplus \bar{x}} = \overline{x \oplus x \oplus x \oplus x \oplus x \oplus \bar{x}} = \overline{x \oplus g}$ . Substituting  $\overline{x \oplus g}$  for  $\overline{x \oplus \bar{x}}$  in **ProofLine[11]'** yields

```
In[44]:= ProofLine[11]'' = ProofLine[11]' /. { $\overline{x \oplus \bar{x}} \rightarrow \overline{x \oplus g}$ }
Out[44]=  $\overline{y \oplus \bar{x} \oplus g \oplus \bar{x} \oplus y \oplus g}$  == y
```

**ProofLine[11]''** is the following substitution instance of line 1.

```
In[45]:= ProofLine[11]''' == (ProofLine[1] /. {y -> y, x -> x ⊕ g})
Out[45]= True
```

### ■ Line 12 of the EQP Proof

Line 12 of the EQP proof is

```
In[46]:= ProofLine[12] =  $\overline{x \oplus \bar{x} \oplus x \oplus x \oplus x \oplus x \oplus x \oplus \bar{x}}$  == x
Out[46]=  $\overline{x \oplus \bar{x} \oplus x \oplus x \oplus x \oplus x \oplus x \oplus \bar{x}}$  == x
```

The EQP proof says only that line 12 follows from lines 1 and 10. To see how, let  $h = \overline{x \oplus x \oplus x \oplus x \oplus \bar{x}}$ . Then, **ProofLine[12]** becomes

```
In[47]:= ProofLine[12]' = ProofLine[12] /. { $\overline{x \oplus x \oplus x \oplus x \oplus \bar{x}} \rightarrow h$ }
Out[47]=  $\overline{x \oplus h \oplus x \oplus \bar{x}}$  == x
```







```
In[60]:= ProofLine[15]'' == (ProofLine[11] //. {y -> k})
Out[60]= True
```

### ■ Line 16 of the EQP Proof—The last line of the proof (finally!)

Line 16 of the EQP proof is

```
In[61]:= ProofLine[16] = x ⊕ x ⊕ x ⊕ x ⊕ x ⊕ x ⊕ x ⊕ x == x ⊕ x ⊕ x ⊕ x ⊕ x
Out[61]= x ⊕ x ⊕ x ⊕ x ⊕ x ⊕ x ⊕ x ⊕ x == x ⊕ x ⊕ x ⊕ x ⊕ x
```

The EQP proof says only that line 16 can be obtained using some substitution strategy involving lines 4, 14, and 15. To see the relevant strategy, let  $\mathbb{1} = x \oplus x \oplus x \oplus \overline{x \oplus \overline{x}}$ . Then, **ProofLine[16]** becomes

```
In[62]:= ProofLine[16]' = ProofLine[16] //. {x ⊕ x ⊕ x ⊕ x ⊕ x → 1}
Out[62]= x ⊕ 1 ⊕ x ⊕ x == 1
```

From line 15, we know that  $x \oplus \mathbb{1} \oplus x \oplus \overline{x} = x \oplus x \oplus \overline{\mathbb{1} \oplus x \oplus \mathbb{1} \oplus x \oplus \overline{x}}$ . To see this, perform the following substitution on **ProofLine[15]**

```
In[63]:= ProofLine[15] //. {x ⊕ x ⊕ x ⊕ x ⊕ x → 1}
Out[63]= x ⊕ x ⊕ 1 ⊕ x ⊕ 1 ⊕ x ⊕ x == x ⊕ 1 ⊕ x ⊕ x
```

Substituting  $x \oplus x \oplus \overline{\mathbb{1} \oplus x \oplus \mathbb{1} \oplus x \oplus \overline{x}}$  for  $x \oplus \mathbb{1} \oplus x \oplus \overline{x}$  in **ProofLine[16]'** yields

```
In[64]:= ProofLine[16]'' = ProofLine[16]' /. {x ⊕ 1 ⊕ x ⊕ x → x ⊕ x ⊕ 1 ⊕ x ⊕ 1 ⊕ x ⊕ x}
Out[64]= x ⊕ x ⊕ 1 ⊕ x ⊕ 1 ⊕ x ⊕ x == 1
```

From line 4, we know that  $x = x \oplus \overline{x} \oplus x \oplus x \oplus x \oplus x \oplus \overline{x} = \overline{\mathbb{1} \oplus x \oplus \overline{x}}$ . Substituting  $\overline{\mathbb{1} \oplus x \oplus \overline{x}}$  for the first instance of  $x$  in **ProofLine[16]''** yields

```
In[65]:= ProofLine[16]''' = 1 ⊕ x ⊕ x ⊕ x ⊕ 1 ⊕ x ⊕ 1 ⊕ x ⊕ x == 1
Out[65]= 1 ⊕ x ⊕ x ⊕ x ⊕ 1 ⊕ x ⊕ 1 ⊕ x ⊕ x == 1
```

Finally, **ProofLine[16]'''** is the following substitution instance of line 14.

```
In[66]:= ProofLine[16]'''' == (ProofLine[14] //. {y -> 1, x ⊕ x ⊕ x ⊕ x ⊕ x → 1})
Out[66]= True
```

## ■ What have we just proved?

We have just proved that  $\overline{x \oplus x \oplus x \oplus x \oplus x \oplus \bar{x} \oplus x \oplus \bar{x}} = x \oplus x \oplus x \oplus x \oplus \bar{x}$ . To see that the *Winker Condition* follows from this, let  $C = x \oplus x \oplus x \oplus \bar{x}$ , and  $D = x \oplus x \oplus \bar{x}$ . Then, it is straightforward to verify that line 16 entails  $\overline{C \oplus D} = \bar{C}$ , which implies the *Winker Condition*.

```
In[67]:= ProofLine[16] //. {x⊕x⊕x⊕x⊕x⊕x̄⊕x⊕x̄→C, x⊕x⊕x̄→D}
```

```
Out[67]= C⊕D==C̄
```

This concludes our *Mathematica*-aided deciphering of the **EQP** proof of the Winker Condition. In conjunction with the results previously reported in Winker (1992), this proof establishes that *all Robbins algebras are Boolean algebras (i.e., that the Robbins conjecture is correct)*.  $\square$

## ■ Concluding Remarks

Although the **EQP** proof of the Robbins conjecture (which has now been verified both automatically and by hand by several different researchers) firmly establishes the *truth* of the Robbins conjecture, the **EQP** proof object does little to help human beings *understand how to prove* the Robbins conjecture. In fact, the **EQP** proof object is quite difficult to follow (or even parse!). After several frustrating and fruitless weeks of trying to work through the **EQP** proof object by hand, I started experimenting with *Mathematica*. Just a couple of days later, I had a complete *Mathematica* reconstruction of the proof! The keys to my success were (1) *Mathematica*'s two-dimensional Boolean notation for the complementation operator  $\bar{\phantom{x}}$  which allowed me to see many complex syntactical patterns that were all but invisible in **EQP**'s one-dimensional notation, and (2) *Mathematica*'s powerful, interactive symbolic engine, which allowed me to experiment with many complicated substitution strategies in "real time." I used *Mathematica* exclusively in both my scratch work, and in generating the polished reconstruction that appears here.

One of the most pressing current problems facing researchers in automated theorem-proving (according to Professor Kenneth Kunen, personal communication) is the translation and reconstruction of complex, unintuitive computer proofs into forms that are more readily understood by human beings. I think my "Robbins success story" illustrates how *Mathematica* can be used quite effectively to help people comprehend complicated computer proofs of this kind.

## ■ Appendix

### ■ All Boolean algebras are Robbins algebras

I mentioned above that, in the context of Boolean algebra, the Robbins equation is *logically equivalent* to the Huntington equation. We can prove this rather easily using *Mathematica*, as follows. In the context of Boolean algebra (specifically, in Boolean *logic*), the binary relation  $\oplus$  is just Boolean disjunction  $\vee$ , and the unary relation  $\bar{\phantom{x}}$  is just Boolean negation  $\neg$ . So, we define the following list of replacement rules to capture the assumption that we are working within Boolean logic.

```
In[68]:= Boolean = {x̄→¬x, x⊕y→x∨y};
```

Then, we use *Mathematica*'s `LogicalExpand` function to construct the following one-line proof of the logical equivalence of the Huntington equation and the Robbins equation, within the context of Boolean algebra.

```
In[69]:= (Overline[y Overline[x] Overline[x] Overline[y] => Overline[x Overline[y] Overline[x] Overline[y] ]]) && (Overline[x Overline[y] Overline[x] Overline[y] => Overline[y Overline[x] Overline[x] Overline[y] ]) // . Boolean
// LogicalExpand
```

```
Out[69]= True
```

Together with the results in Huntington (1933a,b), this entails that *all Boolean algebras are Robbins algebras*.  $\square$

## ■ Double Negation is sufficient to make any Robbins algebra Boolean

Earlier, I claimed that the *double negation* condition  $\overline{\overline{x}} = x$  is sufficient to make any Robbins algebra Boolean. We can demonstrate this using *Mathematica*, as follows.

The Robbins equation says

```
In[70]:= Robbins = (RobbinsExpand[x, y] == x)
```

```
Out[70]= Overline[x Overline[y] Overline[x] Overline[y] ] == x
```

And, the Huntington equation says

```
In[71]:= Huntington = (Overline[x Overline[y] Overline[x] Overline[y] ] == x)
```

```
Out[71]= Overline[y Overline[x] Overline[x] Overline[y] ] == x
```

If we assume *double negation*, then the Robbins equation is equivalent to

```
In[72]:= (RobbinsExpand[Overline[x], y] == x) // . {Overline[Overline[x]] -> x}
```

```
Out[72]= Overline[y Overline[x] Overline[x] Overline[y] ] == x
```

which is obviously identical to the Huntington equation.  $\square$

## ■ Acknowledgments

I would like to thank Tina Eliassi–Rad and Professor Kenneth Kunen for getting me interested in the Robbins problem, Bill McCune for solving the Robbins problem and for inspiring me to publish my *Mathematica* reconstruction of his **EQP** proof object, and an anonymous referee of *Mathematica in Education and Research* for his detailed comments and suggestions on an earlier version of this paper.

## ■ Notes

<sup>1</sup> Much of the history of the Robbins problem reported here is taken from McCune (1997) and Wos *et al* (1992). See McCune (1997) for the full story on the recent computer proof of the Robbins conjecture. Also, check-out Bill McCune's home page at <http://www.mcs.anl.gov/home/mccune/ar/>, and, especially, his "Robbins Algebras are Boolean" page at <http://www.mcs.anl.gov/home/mccune/ar/robbins/> for lots of information about the Robbins conjecture and automated reasoning in general.

<sup>2</sup> See Kneale and Kneale (1962, pages 404–427) for a scholarly trace of the historical development of Boolean algebra and Boolean logic from Boole (1854) to Huntington (1933 a,b).

<sup>3</sup> The relation  $\otimes$  is then defined in terms of  $\oplus$  and  $\bar{\cdot}$  as follows:  $x \otimes y \equiv \overline{\bar{x} \oplus \bar{y}}$ . Showing that Huntington's simpler (1933a,b) axiomatization is equivalent to his more complex (1904) axiomatization is not very difficult to do by hand. Halmos (1963, page 9) gives this proof as an introductory exercise.

<sup>4</sup> See McCune (1997) for all technical details, and see Kolata (1996), and Peterson (1997) for less technical news reports about the computer proof and its significance.

<sup>5</sup> See Argonne's automated reasoning page at <http://www.mcs.anl.gov/home/mccune/ar/> for more about **EQP** and **Otter**. See, also, Wos *et al* (1992) for a general introduction to the theory and practice of automated reasoning.

<sup>6</sup> Apparently, Boole (1854, page 147) was the first to use the overbar notation  $\bar{x}$  to denote the negation or complementation of  $x$  in the context of formal logic and algebra. This notation was later adopted by C.S. Peirce, but not by E.V. Huntington. See Cajori (1993, ¶ 685) for the historical development of the overbar notation in formal logic and algebra.

<sup>7</sup> The actual **EQP** output of this proof can be obtained from the following URL <http://www.mcs.anl.gov/home/mccune/ar/robbins/eqp-theorem.proof.txt>.

<sup>8</sup> We don't tell *Mathematica* anything about the attributes of the complementation operator  $\bar{\cdot}$ . This is as it should be. We are only allowed to assume that  $\bar{\cdot}$  is a unary function obeying the Robbins equation.

<sup>9</sup> **EQP** also handles commutativity and associativity *implicitly* in this way. No information is given in the **EQP** proof object about the roles that commutativity and associativity play in the proof.

## ■ References

Boole, G., 1854, *The Laws of Thought*, Dover, New York.

Cajori, F., 1933, *A History of Mathematical Notations*, Dover, New York.

Halmos, P., 1963, *Lectures on Boolean Algebras*, Van Nostrand, New York.

Huntington, E.V., 1904, Sets of independent postulates for the algebra of logic, *Transactions of the American Mathematical Society* **5**, 288–309.

Huntington, E.V., 1933a, New sets of independent postulates for the algebra of logic, *Transactions of the American Mathematical Society* **35**, 274–304.

Huntington, E.V., 1933b, Boolean algebra: A correction, *Transactions of the American Mathematical Society* **35**, 557–558.

Kneale, W., and Kneale, M., 1962, *The Development of Logic*, Clarendon: Oxford University Press, Oxford.

Kolata, G., 1996, Computer Math Proof Shows Reasoning Power, *The New York Times*, December 10, 1996.

McCune, W., 1997, Solution of the Robbins Problem, forthcoming in the *Journal of Automated Reasoning*. A preprint of this paper is available at <ftp://info.mcs.anl.gov/pub/Otter/www-misc/robbins-jar-final.ps.gz>.

Peterson, I., 1997, Computers and Proof: Applying automated reasoning to prove mathematical theorems, *Science News* **151**(12), March 22, 1997.

Stoll, R., 1963, *Set Theory and Logic*, W.H. Freeman and Company, San Francisco.

Tarski, A., Henkin, L., and Monk, D., 1971, *Cylindric Algebras: Part I*, North-Holland, New York.

Winker, S., 1992, Absorption and idempotency criteria for a problem in near-Boolean algebras, *Journal of Algebra* **153**(2), 414–423.

Wos, L., Overbeek, R., Lusk, E., and Boyle, J., 1992, *Automated Reasoning: Introduction and Applications*, Second Edition, McGraw Hill, New York.

## ■ About the Author

Branden Fitelson is a philosophy graduate student at the University of Wisconsin–Madison. He earned his BS (Mathematics) from the University of Wisconsin–Madison in 1992, and his MA (Philosophy) from the University of Wisconsin–Madison in 1997. He is currently working toward his PhD (Philosophy). His research interests include philosophy of science and logic. See Branden's home page at <http://polyglot.lss.wisc.edu/philosophy/PEOPLE/TA/BF/BF.HTM>, for more professional and personal information about the author.