A Complete Proof of the Robbins Conjecture

Matthew Wampler-Doty

September 19, 2015

Abstract


Contents

1 Robbins Conjecture 1

2 Axiom Systems 1

2.1 Common Algebras 2

2.2 Boolean Algebra 2

2.3 Huntington’s Algebra 3

2.4 Robbins’ Algebra 3

3 Equivalence 3

3.1 Boolean Algebra 3

3.2 Huntington Algebra 4

3.3 Robbins’ Algebra 6

1 Robbins Conjecture

theory Robbins-Conjecture
imports Main
begin


2 Axiom Systems

The following presents several axiom systems that shall be under study.
The first axiom sets common systems that underly all of the systems we shall be looking at.

The second system is a reformulation of Boolean algebra. We shall follow pages 7–8 in S. Koppelberg, *General Theory of Boolean Algebras*, Volume 1 of *Handbook of Boolean Algebras*. North Holland, 1989. Note that our formulation deviates slightly from this, as we only provide one distribution axiom, as the dual is redundant.

The third system is Huntington’s algebra and the fourth system is Robbins’ algebra.

Apart from the common system, all of these systems are demonstrated to be equivalent to the library formulation of Boolean algebra, under appropriate interpretation.

### 2.1 Common Algebras

```haskell
class common-algebra = uminus +
fixes inf :: 'a ⇒ 'a ⇒ 'a (infixl ⊓ 70)
fixes sup :: 'a ⇒ 'a ⇒ 'a (infixl ⊔ 65)
fixes bot :: 'a (⊥)
fixes top :: 'a (⊤)
assumes sup-assoc: x ⊔ (y ⊔ z) = (x ⊔ y) ⊔ z
assumes sup-comm: x ⊔ y = y ⊔ x
context common-algebra begin
  definition less-eq :: 'a ⇒ 'a ⇒ bool (infix ⊑ 50) where
    x ⊑ y = (x ⊔ y = y)
  definition less :: 'a ⇒ 'a ⇒ bool (infix ⊏ 50) where
    x ⊏ y = (x ⊑ y ∧ ¬ y ⊑ x)
  definition minus :: 'a ⇒ 'a ⇒ 'a (infixl − 65) where
    minus x y = (x ⊓ − y)
  definition secret-object1 :: 'a (ι) where
    ι = (SOME x. True)
end
```

### 2.2 Boolean Algebra

```haskell
class boolean-algebra-II =
  common-algebra +
  assumes inf-comm: x ⊓ y = y ⊓ x
```
assumes inf-assoc: \( x \cap (y \cap z) = (x \cap y) \cap z \)
assumes sup-absorb: \( x \cup (x \cap y) = x \)
assumes inf-absorb: \( x \cap (x \cup y) = x \)
assumes sup-inf-distrib1: \( x \cup y \cap z = (x \cup y) \cap (x \cup z) \)
assumes sup-compl: \( x \cup -x = \top \)
assumes inf-compl: \( x \cap -x = \bot \)

2.3 Huntington’s Algebra

class huntington-algebra = ext-common-algebra +
assumes huntington: \(-(-x \cup -y) \cup -(-x \cup y) = x\)

2.4 Robbins’ Algebra

class robbins-algebra = ext-common-algebra +
assumes robbins: \(-((-x \cup y) \cup -(x \cup -y)) = x\)

3 Equivalence
With our axiom systems defined, we turn to providing equivalence results between them.

We shall begin by illustrating equivalence for our formulation and the library formulation of Boolean algebra.

3.1 Boolean Algebra
The following provides the canonical definitions for order and relative complementation for Boolean algebras. These are necessary since the Boolean algebras presented in the Isabelle/HOL library have a lot of structure, while our formulation is considerably simpler.

Since our formulation of Boolean algebras is considerably simple, it is easy to show that the library instantiates our axioms.

context boolean-algebra-II begin

lemma boolean-II-is-boolean:
  class boolean-algebra-II minus uminus (op \cap) (op \sqcup) (op \sqsubseteq) (op \sqsupseteq) \bot \top
  ⟨proof⟩
end

context boolean-algebra begin

lemma boolean-is-boolean-II:
  class boolean-algebra minus uminus inf sup bot top
  ⟨proof⟩
end
3.2 Huntington Algebra

We shall illustrate here that all Boolean algebra using our formulation are Huntington algebras, and illustrate that every Huntington algebra may be interpreted as a Boolean algebra.

Since the Isabelle/HOL library has good automation, it is convenient to first show that the library instances Huntington algebras to exploit previous results, and then use our previously derived correspondence.

context boolean-algebra begin

lemma boolean-is-huntington:
  class.huntington-algebra uminus inf sup bot top
⟨proof⟩
end

context boolean-algebra-II begin

lemma boolean-II-is-huntington:
  class.huntington-algebra uminus (op ⊓) (op ⊔) ⊥⊤
⟨proof⟩
end

context huntington-algebra begin

lemma huntington-id:
  x ⊓¬x = ¬x ⊔(¬x)
⟨proof⟩

lemma dbl-neg:
  ¬(¬x) = x
⟨proof⟩

lemma towards-sup-compl:
  x ⊔¬x = y ⊔¬y
⟨proof⟩

lemma sup-compl:
  x ⊔¬x = ⊤
⟨proof⟩

lemma towards-inf-compl:
  x ⊓¬x = y ⊓¬y
⟨proof⟩

lemma inf-compl:
  x ⊓¬x = ⊥
⟨proof⟩

lemma towards-idem:
  ⊥ = ⊥ ⊔⊥
⟨proof⟩

lemma sup-ident:
  x ⊔⊥ = x
⟨proof⟩

4
lemma inf-ident: \( x \cap \top = x \) 
(\textit{proof})

lemma sup-idem: \( x \cup x = x \) 
(\textit{proof})

lemma inf-idem: \( x \cap x = x \) 
(\textit{proof})

lemma sup-nil: \( x \cup \top = \top \) 
(\textit{proof})

lemma inf-nil: \( x \cap \bot = \bot \) 
(\textit{proof})

lemma sup-absorb: \( x \cup x \cap y = x \) 
(\textit{proof})

lemma inf-absorb: \( x \cap (x \cup y) = x \) 
(\textit{proof})

lemma partition: \( x \cap y \cup x \cap -y = x \) 
(\textit{proof})

lemma demorgans1: \( -(x \cap y) = -x \cup -y \) 
(\textit{proof})

lemma demorgans2: \( -(x \cup y) = -x \cap -y \) 
(\textit{proof})

lemma inf-comm: \( x \cap y = y \cap x \) 
(\textit{proof})

lemma inf-assoc: \( x \cap (y \cap z) = x \cap y \cap z \) 
(\textit{proof})

lemma inf-sup-distrib1: \( x \cap (y \cup z) = (x \cap y) \cup (x \cap z) \) 
(\textit{proof})

lemma sup-inf-distrib1: 
\[ x \cup (y \cap z) = (x \cup y) \cap (x \cup z) \] 
(\textit{proof})

lemma huntington-is-boolean-II: 
\[ \text{class.boolean-algebra-II aminus (op \cap) (op \cup) \bot \top} \] 
(\textit{proof})

lemma huntington-is-boolean: 
\[ \text{class.boolean-algebra minus aminus (op \cap) (op \sqsubseteq) (op \sqcup) (op \cup) \bot \top} \]
3.3 Robbins’ Algebra

context boolean-algebra begin
lemma boolean-is-robbins:
  class.robbins-algebra uminus inf sup bot top
⟨proof⟩
end

context boolean-algebra-II begin
lemma boolean-II-is-robbins:
  class.robbins-algebra uminus inf sup bot top
⟨proof⟩
end

context huntington-algebra begin
lemma huntington-is-robbins:
  class.robbins-algebra uminus inf sup bot top
⟨proof⟩
end

Before diving into the proof that the Robbins algebra is Boolean, we shall present some shorthand machinery

context common-algebra begin

primrec copyp :: nat ⇒ 'a ⇒ 'a (infix ⊗ 80)
where
  copyp-0: 0 ⊗ x = x
| copyp-Suc: (Suc k) ⊗ x = (k ⊗ x) ⊔ x

no-notation
  Product-Type.Times (infixr × 80)

primrec copy :: nat ⇒ 'a ⇒ 'a (infix × 85)
where
  0 × x = x
| (Suc k) × x = k ⊗ x

lemma one: 1 × x = x
⟨proof⟩

lemma two: 2 × x = x ⊔ x
⟨proof⟩
lemma three: 3 × x = x ∪ x ∪ x
⟨proof⟩

lemma four: 4 × x = x ∪ x ∪ x ∪ x
⟨proof⟩

lemma five: 5 × x = x ∪ x ∪ x ∪ x ∪ x
⟨proof⟩

lemma six: 6 × x = x ∪ x ∪ x ∪ x ∪ x ∪ x
⟨proof⟩

lemma copyp-distrib: k ⊗ (x ∪ y) = (k ⊗ x) ∪ (k ⊗ y)
⟨proof⟩

corollary copy-distrib: k × (x ∪ y) = (k × x) ∪ (k × y)
⟨proof⟩

lemma copyp-arith: (k + l + 1) ⊗ x = (k × x) ∪ (l × x)
⟨proof⟩

lemma copy-arith:
   assumes k ≠ 0 and l ≠ 0
   shows (k + l) × x = (k × x) ∪ (l × x)
⟨proof⟩

end

The theorem asserting all Robbins algebras are Boolean comes in 6 movements.
   First: The Winker identity is proved.
   Second: Idempotence for a particular object is proved. Note that falsum
   is defined in terms of this object.
   Third: An identity law for falsum is derived.
   Fourth: Idempotence for supremum is derived.
   Fifth: The double negation law is proven
   Sixth: Robbin's algebras are proven to be Huntington Algebras.

context robbins-algebra begin

definition secret-object2 :: 'a (α) where
   α = −(−(i ∪ i ∪ i) ∪ i)
definition secret-object3 :: 'a (β) where
   β = i ∪ i
definition secret-object4 :: 'a (δ) where
   δ = β ∪ (−(α ∪ −β) ∪ −(α ∪ −β))
**Definition** secret-object5 :: 'a (\(\gamma\)) where
\[ \gamma = \delta \cup -(\delta \cup -\delta) \]

**Definition** winker-object :: 'a (\(\varrho\)) where
\[ \varrho = \gamma \cup \gamma \cup \gamma \]

**Definition** fake-bot :: 'a (\(\bot\)) where
\[ \bot = -(\varrho \cup -\varrho) \]

**Lemma** robbins2: \(y = -(-(x \cup y) \cup -(x \cup y))\)

**Lemma** mann0: \(-(x \cup y) = -(-(x \cup y) \cup -x \cup y) \cup y\)

**Lemma** mann1: \(-(x \cup y) = -(-(x \cup y) \cup x \cup y) \cup y\)

**Lemma** mann2: \(y = -(-(x \cup y) \cup x \cup y \cup y) \cup -(-x \cup y)\)

**Lemma** mann3: \(z = -(-(x \cup y) \cup x \cup y \cup y) \cup -(-x \cup y) \cup -(y \cup z) \cup z) \cup -(y \cup z)\)

**Lemma** mann4: \(-(y \cup z) =
-(-(y \cup z) \cup x \cup y \cup y) \cup -(x \cup y) \cup -(-x \cup y) \cup -(y \cup z) \cup z) \cup z)\)

**Lemma** mann5: \(u =
-(-(x \cup y) \cup x \cup y \cup y) \cup
-(-x \cup y) \cup -y \cup z \cup z \cup u \cup
-(-y \cup z) \cup u)\)

**Lemma** mann6: 
\(-3 \times x \cup x) = -(-(3 \times x \cup x) \cup -3 \times x) \cup -(-(3 \times x \cup x) \cup 5 \times x)\)

**Lemma** mann7: 
\(-3 \times x = -(-(3 \times x \cup x) \cup 5 \times x)\)

**Lemma** mann8: 
\(-3 \times x \cup x) \cup 2 \times x = -(-(3 \times x \cup x) \cup -3 \times x \cup 2 \times x) \cup -3 \times x)\)

(is ?lhs = ?rhs)


lemma mann9: \( x = -(-3x \sqcup x) \sqcup -3x \)  
\( \langle \text{proof} \rangle \)

lemma mann10: \( y = -(-(-3x \sqcup x) \sqcup -3x \sqcup y) \sqcup -(x \sqcup y) \)  
\( \langle \text{proof} \rangle \)

theorem mann: \( 2x = -(-3x \sqcup x) \sqcup 2x \)  
\( \langle \text{proof} \rangle \)

corollary winkerr: \( \alpha \sqcup \beta = \beta \)  
\( \langle \text{proof} \rangle \)

corollary winker: \( \beta \sqcup \alpha = \beta \)  
\( \langle \text{proof} \rangle \)

corollary multi-winker: \( \beta \sqcup k \otimes \alpha = \beta \)  
\( \langle \text{proof} \rangle \)

corollary multi-winker: \( \beta \sqcup k \times \alpha = \beta \)  
\( \langle \text{proof} \rangle \)

lemma less-eq-introp:  
\(- (x \sqcup -(y \sqcup z)) = -(x \sqcup y \sqcup -z) \implies y \sqsubseteq x \)  
\( \langle \text{proof} \rangle \)

corollary less-eq-intro:  
\(- (x \sqcup -(y \sqcup z)) = -(x \sqcup y \sqcup -z) \implies x \sqcup y = x \)  
\( \langle \text{proof} \rangle \)

lemma eq-intro:  
\(- (x \sqcup -(y \sqcup z)) = -(y \sqcup -(x \sqcup z)) \implies x = y \)  
\( \langle \text{proof} \rangle \)

lemma copyp0:  
assumes \(- (x \sqcup -y) = z \)  
shows \(- (x \sqcup -(y \sqcup k \otimes (x \sqcup z))) = z \)  
\( \langle \text{proof} \rangle \)

lemma copyp1:  
assumes \(- (x \sqcup -y) \sqcup -y = x \)  
shows \(- (y \sqcup k \otimes (x \sqcup -(x \sqcup -y))) = -y \)  
\( \langle \text{proof} \rangle \)

corollary copyp2:  
assumes \(- (x \sqcup y) = -y \)  
shows \(- (y \sqcup k \otimes (x \sqcup -(x \sqcup -y))) = -y \)  
\( \langle \text{proof} \rangle \)
lemma two-threep:  
  assumes \(- (2 \times x \sqcup y) = -y\)  
  and \(- (3 \times x \sqcup y) = -y\)  
  shows \(2 \times x \sqcup y = 3 \times x \sqcup y\)  

lemma two-three:  
  assumes \(- (x \sqcup y) = -y \lor -(x \sqcup -y) \sqcup -y) = x\)  
  shows \(y \sqcup 2 \times (x \sqcup -(x \sqcup -y)) = y \sqcup 3 \times (x \sqcup -(x \sqcup -y))\)  
     \((is \ y \sqcup \?z2 = y \sqcup \?z3)\)  

lemma sup-idem: \(q \sqcup q = q\)  

lemma sup-ident: \(x \sqcup \bot \bot = x\)  

lemma dbl-neg: \(-(-x) = x\)  

theorem robbins-is-huntington:  
  class.huntington-algebra uminus (op \sqcap) (op \sqcup) \bot \top  

theorem robbins-is-boolean-II:  
  class.boolean-algebra-II uminus (op \sqcap) (op \sqcup) \bot \top  

theorem robbins-is-boolean:  
  class.boolean-algebra minus uminus (op \sqcap) (op \sqsubseteq) (op \sqsupseteq) (op \sqcup) \bot \top  

end  

no-notation secret-object1 (ι)  
  and secret-object2 (α)  
  and secret-object3 (β)  
  and secret-object4 (δ)  
  and secret-object5 (γ)  
  and winker-object (ϱ)
\begin{verbatim}
and \ less-eq  (infix \sqsubseteq 50)
and \ less       (infix \sqsubseteq 50)
and \ inf       (infixl \sqcap 70)
and \ sup       (infixl \sqcup 65)
and \ top       (\perp)
and \ bot       (\bot)
and \ copyp     (infix \otimes 80)
and \ copy      (infix \times 85)

notation
  \textit{Product-Type.Times}  (infixr \times 80)

end
\end{verbatim}