A Complete Proof of the Robbins Conjecture

Matthew Wampler-Doty

March 17, 2025

Abstract

The document gives a formalization of the proof of the Robbins conjecture, following A. Mann, *A Complete Proof of the Robbins Conjecture*, 2003.

Contents

1 Robbins Conjecture

theory *Robbins-Conjecture*
imports *Main*
begin

The document gives a formalization of the proof of the Robbins conjecture, following A. Mann, *A Complete Proof of the Robbins Conjecture*, 2003, DOI 10.1.1.6.7838

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

class *common-algebra* = *uminus* +
  fixes *inf* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** ‹⊓› *70*)
  fixes *sup* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** ‹⊔› *65*)
  fixes *bot* :: $'a$ (‹⊥›)
  fixes *top* :: $'a$ (‹⊤›)
  assumes *sup-assoc*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
  assumes *sup-comm*: $x \sqcup y = y \sqcup x$

context *common-algebra* **begin**

definition *less-eq* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹⊑› *50*) **where**
  $x \sqsubseteq y = (x \sqcup y = y)$
definition *less* :: $'a \Rightarrow 'a \Rightarrow bool$ (**infix** ‹⊏› *50*) **where**
  $x \sqsubset y = (x \sqsubseteq y \land \neg\, y \sqsubseteq x)$
definition *minus* :: $'a \Rightarrow 'a \Rightarrow 'a$ (**infixl** ‹−› *65*) **where**
  $minus\ x\ y = (x \sqcap -\ y)$

definition *secret-object1* :: $'a$ (‹ι›) **where**
  $\iota = (SOME\ x.\ True)$

end

class *ext-common-algebra* = *common-algebra* +
  assumes *inf-eq*: $x \sqcap y = -(-\ x \sqcup -\ y)$
  assumes *top-eq*: $\top = \iota \sqcup -\ \iota$
  assumes *bot-eq*: $\bot = -(\iota \sqcup -\ \iota)$

2.2 Boolean Algebra

class *boolean-algebra-II* =
  common-algebra +
  assumes *inf-comm*: $x \sqcap y = y \sqcap x$

assumes *inf-assoc*: $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$
assumes *sup-absorb*: $x \sqcup (x \sqcap y) = x$
assumes *inf-absorb*: $x \sqcap (x \sqcup y) = x$
assumes *sup-inf-distrib1*: $x \sqcup y \sqcap z = (x \sqcup y) \sqcap (x \sqcup z)$
assumes *sup-compl*: $x \sqcup - x = \top$
assumes *inf-compl*: $x \sqcap - x = \bot$

2.3 Huntington's Algebra

class *huntington-algebra* = *ext-common-algebra* +
assumes *huntington*: $- (-x \sqcup -y) \sqcup - (-x \sqcup y) = x$

2.4 Robbins' Algebra

class *robbins-algebra* = *ext-common-algebra* +
assumes *robbins*: $- (- (x \sqcup y) \sqcup - (x \sqcup -y)) = x$

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 minus uminus $(\sqcap)$ $(\sqsubseteq)$ $(\sqsubset)$ $(\sqcup)$ $\bot$ $\top$
⟨*proof*⟩

end

context *boolean-algebra* **begin**

lemma *boolean-is-boolean-II*:
class.boolean-algebra-II 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 $(\sqcap)$ $(\sqcup)$ $\bot$ $\top$
⟨*proof*⟩

end

context *huntington-algebra* **begin**

lemma *huntington-id*: $x \sqcup -x = -x \sqcup -(-x)$
⟨*proof*⟩

lemma *dbl-neg*: $-\ (-x) = x$
⟨*proof*⟩

lemma *towards-sup-compl*: $x \sqcup -x = y \sqcup -y$
⟨*proof*⟩

lemma *sup-compl*: $x \sqcup -x = \top$
⟨*proof*⟩

lemma *towards-inf-compl*: $x \sqcap -x = y \sqcap -y$
⟨*proof*⟩

lemma *inf-compl*: $x \sqcap -x = \bot$
⟨*proof*⟩

lemma *towards-idem*: $\bot = \bot \sqcup \bot$
⟨*proof*⟩

lemma *sup-ident*: $x \sqcup \bot = x$
⟨*proof*⟩

lemma *inf-ident*: $x \sqcap \top = x$
⟨*proof*⟩

lemma *sup-idem*: $x \sqcup x = x$
⟨*proof*⟩

lemma *inf-idem*: $x \sqcap x = x$
⟨*proof*⟩

lemma *sup-nil*: $x \sqcup \top = \top$
⟨*proof*⟩

lemma *inf-nil*: $x \sqcap \bot = \bot$
⟨*proof*⟩

lemma *sup-absorb*: $x \sqcup x \sqcap y = x$
⟨*proof*⟩

lemma *inf-absorb*: $x \sqcap (x \sqcup y) = x$
⟨*proof*⟩

lemma *partition*: $x \sqcap y \sqcup x \sqcap -y = x$
⟨*proof*⟩

lemma *demorgans1*: $-(x \sqcap y) = -x \sqcup -y$
⟨*proof*⟩

lemma *demorgans2*: $-(x \sqcup y) = -x \sqcap -y$
⟨*proof*⟩

lemma *inf-comm*: $x \sqcap y = y \sqcap x$
⟨*proof*⟩

lemma *inf-assoc*: $x \sqcap (y \sqcap z) = x \sqcap y \sqcap z$
⟨*proof*⟩

lemma *inf-sup-distrib1*: $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$
⟨*proof*⟩

lemma *sup-inf-distrib1*:
$x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$
⟨*proof*⟩

lemma *huntington-is-boolean-II*:
class.boolean-algebra-II uminus $(\sqcap)$ $(\sqcup)$ $\bot$ $\top$
⟨*proof*⟩

lemma *huntington-is-boolean*:
class.boolean-algebra minus uminus $(\sqcap)$ $(\sqsubseteq)$ $(\sqsubset)$ $(\sqcup)$ $\bot$ $\top$

⟨*proof*⟩
end

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* :: $nat \Rightarrow 'a \Rightarrow 'a$ (**infix** ‹⊗› *80*)
where
copyp-0: $0 \otimes x = x$
| *copyp-Suc*: $(Suc\ k) \otimes x = (k \otimes x) \sqcup x$

unbundle *no set-product-syntax*

primrec *copy* :: $nat \Rightarrow 'a \Rightarrow 'a$ (**infix** ‹×› *85*)
where
$0 \times x = x$
| $(Suc\ k) \times x = k \otimes x$

lemma *one*: $1 \times x = x$
⟨*proof*⟩

lemma *two*: $2 \times x = x \sqcup x$
⟨*proof*⟩

lemma *three*: $3 \times x = x \sqcup x \sqcup x$
⟨*proof*⟩

lemma *four*: $4 \times x = x \sqcup x \sqcup x \sqcup x$
⟨*proof*⟩

lemma *five*: $5 \times x = x \sqcup x \sqcup x \sqcup x \sqcup x$
⟨*proof*⟩

lemma *six*: $6 \times x = x \sqcup x \sqcup x \sqcup x \sqcup x \sqcup x$
⟨*proof*⟩

lemma *copyp-distrib*: $k \otimes (x \sqcup y) = (k \otimes x) \sqcup (k \otimes y)$
⟨*proof*⟩

corollary *copy-distrib*: $k \times (x \sqcup y) = (k \times x) \sqcup (k \times y)$
⟨*proof*⟩

lemma *copyp-arith*: $(k + l + 1) \otimes x = (k \otimes x) \sqcup (l \otimes x)$
⟨*proof*⟩

lemma *copy-arith*:
  assumes $k \neq 0$ **and** $l \neq 0$
    shows $(k + l) \times x = (k \times x) \sqcup (l \times x)$
⟨*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* :: $'a$ ($‹\alpha›$) **where**
  $\alpha = -(-(\iota \sqcup \iota \sqcup \iota) \sqcup \iota)$
definition *secret-object3* :: $'a$ ($‹\beta›$) **where**
  $\beta = \iota \sqcup \iota$
definition *secret-object4* :: $'a$ ($‹\delta›$) **where**
  $\delta = \beta \sqcup (-(\alpha \sqcup -\beta) \sqcup -(\alpha \sqcup -\beta))$
definition *secret-object5* :: $'a$ ($‹\gamma›$) **where**

$\gamma = \delta \sqcup -(\delta \sqcup -\delta)$
definition *winker-object* :: $'a$ ($\langle\varrho\rangle$) **where**
$\varrho = \gamma \sqcup \gamma \sqcup \gamma$
definition *fake-bot* :: $'a$ ($\langle\bot\bot\rangle$) **where**
$\bot\bot = -(\varrho \sqcup -\varrho)$

lemma *robbins2*: $y = -(-(-x \sqcup y) \sqcup -(x \sqcup y))$
⟨*proof*⟩

lemma *mann0*: $-(x \sqcup y) = -(-(-(x \sqcup y) \sqcup -x \sqcup y) \sqcup y)$
⟨*proof*⟩

lemma *mann1*: $-(-x \sqcup y) = -(-(-(-x \sqcup y) \sqcup x \sqcup y) \sqcup y)$
⟨*proof*⟩

lemma *mann2*: $y = -(-(-(-x \sqcup y) \sqcup x \sqcup y \sqcup y) \sqcup -(-x \sqcup y))$
⟨*proof*⟩

lemma *mann3*: $z = -(-(-(-(-x \sqcup y) \sqcup x \sqcup y \sqcup y) \sqcup -(-x \sqcup y) \sqcup z) \sqcup -(y \sqcup z))$
⟨*proof*⟩

lemma *mann4*: $-(y \sqcup z) =$
$-(-(-(-(-x \sqcup y) \sqcup x \sqcup y \sqcup y) \sqcup -(-x \sqcup y) \sqcup -(y \sqcup z) \sqcup z) \sqcup z)$
⟨*proof*⟩

lemma *mann5*: $u =$
$-(-(-(-(-(-x \sqcup y) \sqcup x \sqcup y \sqcup y) \sqcup$
$-(-x \sqcup y) \sqcup - (y \sqcup z) \sqcup z) \sqcup z \sqcup u) \sqcup$
$-(-(y \sqcup z) \sqcup u))$
⟨*proof*⟩

lemma *mann6*:
$-(- 3\times x \sqcup x) = -(-(-(- 3\times x \sqcup x) \sqcup - 3\times x) \sqcup -(-(- 3\times x \sqcup x) \sqcup 5\times x))$
⟨*proof*⟩

lemma *mann7*:
$- 3\times x = -(-(- 3\times x \sqcup x) \sqcup 5\times x)$
⟨*proof*⟩

lemma *mann8*:
$-(- 3\times x \sqcup x) \sqcup 2\times x = -(-(-(- 3\times x \sqcup x) \sqcup - 3\times x \sqcup 2\times x) \sqcup - 3\times x)$
(**is** *?lhs* = *?rhs*)
⟨*proof*⟩

lemma *mann9*: $x = -(-(-\ 3{\times}x \sqcup x) \sqcup -\ 3{\times}x\ )$
⟨*proof*⟩

lemma *mann10*: $y = -(-(-(-\ 3{\times}x \sqcup x) \sqcup -\ 3{\times}x \sqcup y) \sqcup -(x \sqcup y))$
⟨*proof*⟩

theorem *mann*: $2{\times}x = -(-\ 3{\times}x \sqcup x) \sqcup 2{\times}x$
⟨*proof*⟩

corollary *winkerr*: $\alpha \sqcup \beta = \beta$
⟨*proof*⟩

corollary *winker*: $\beta \sqcup \alpha = \beta$
⟨*proof*⟩

corollary *multi-winkerp*: $\beta \sqcup k \otimes \alpha = \beta$
⟨*proof*⟩

corollary *multi-winker*: $\beta \sqcup k \times \alpha = \beta$
⟨*proof*⟩

lemma *less-eq-introp*:
$-(x \sqcup -(y \sqcup z)) = -(x \sqcup y \sqcup -z) \Longrightarrow y \sqsubseteq x$
⟨*proof*⟩

corollary *less-eq-intro*:
$-(x \sqcup -(y \sqcup z)) = -(x \sqcup y \sqcup -z) \Longrightarrow x \sqcup y = x$
⟨*proof*⟩

lemma *eq-intro*:
$-(x \sqcup -(y \sqcup z)) = -(y \sqcup -(x \sqcup z)) \Longrightarrow x = y$
⟨*proof*⟩

lemma *copyp0*:
assumes $-(x \sqcup -y) = z$
shows $-(x \sqcup -(y \sqcup k \otimes (x \sqcup z))) = z$
⟨*proof*⟩

lemma *copyp1*:
assumes $-(-(x \sqcup -y) \sqcup -y) = x$
shows $-(y \sqcup k \otimes (x \sqcup -(x \sqcup -y))) = -y$
⟨*proof*⟩

corollary *copyp2*:
assumes $-(x \sqcup y) = -y$
shows $-(y \sqcup k \otimes (x \sqcup -(x \sqcup -y))) = -y$
⟨*proof*⟩

lemma *two-threep*:
  assumes $-(2 \times x \sqcup y) = -y$
    and $-(3 \times x \sqcup y) = -y$
   shows $2 \times x \sqcup y = 3 \times x \sqcup y$
⟨*proof*⟩

lemma *two-three*:
  assumes $-(x \sqcup y) = -y \vee -(-(x \sqcup -y) \sqcup -y) = x$
   shows $y \sqcup 2 \times (x \sqcup -(x \sqcup -y)) = y \sqcup 3 \times (x \sqcup -(x \sqcup -y))$
     (**is** $y \sqcup ?z2 = y \sqcup ?z3$)
⟨*proof*⟩

lemma *sup-idem*: $\varrho \sqcup \varrho = \varrho$
⟨*proof*⟩

lemma *sup-ident*: $x \sqcup \bot\bot = x$
⟨*proof*⟩

lemma *dbl-neg*: $-(-x) = x$
⟨*proof*⟩

theorem *robbins-is-huntington*:
 class.huntington-algebra uminus (⊓) (⊔) ⊥ ⊤
⟨*proof*⟩

theorem *robbins-is-boolean-II*:
 class.boolean-algebra-II uminus (⊓) (⊔) ⊥ ⊤
⟨*proof*⟩

theorem *robbins-is-boolean*:
 class.boolean-algebra minus uminus (⊓) (⊑) (⊏) (⊔) ⊥ ⊤
⟨*proof*⟩

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›)
  and *less-eq* (**infix** ‹⊑› *50*)

and *less* (**infix** ‹⊏› *50*)
and *inf* (**infixl** ‹⊓› *70*)
and *sup* (**infixl** ‹⊔› *65*)
and *top* (‹⊤›)
and *bot* (‹⊥›)
and *copyp* (**infix** ‹⊗› *80*)
and *copy* (**infix** ‹×› *85*)

notation
Product-Type.Times (**infixr** ‹×› *80*)

end