A Complete Proof of the Robbins Conjecture

Matthew Wampler-Doty

February 23, 2021

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 . . . . . . . . . . . . . . . . . . . . . . . . 9

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

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

class ext-common-algebra = common-algebra +
 assumes inf-eq: x ⊓ y = x ⊓ (x ⊔ y)
 assumes top-eq: ⊤ = ⊤ ⊔ ⊥
 assumes bot-eq: ⊥ = ⊥ ⊔ ⊤

2.2 Boolean Algebra

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: \(-(-x \cup -x) \cup (-x \cup -y) = x\)

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.

class boolean-algebra-II begin

apply unfold-locales
apply (metis inf-assoc inf-assoc inf-assoc inf-comm inf-compl
less-def less-eq-def minus-def
sup-absorb sup-absorb sup-comm
sup-compl sup-inf-distrib1
sup-absorb inf-comm)\
done
end

context boolean-algebra-II begin
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
apply unfold-locale
apply (metis double-compl inf-sup-distrib1 inf-top-right
  compl-inf inf-commute inf-sup-absorb compl-sup sup-compl-top
  sup-compl-top sup-assoc)+
done
end

context boolean-algebra-II begin
lemma boolean-II-is-huntington:
  class.huntington-algebra-II uminus (⊓) (⊔) ⊥ ⊤
proof –
  interpret boolean:
    boolean-algebra minus uminus (⊓) (⊔) ⊥ ⊤
  by (fact boolean-II-is-boolean)
  show ?thesis by (simp add: boolean.boolean-is-huntington)
qed
end

context huntington-algebra begin
lemma huntington-id: x ⊔ ¬x = ¬x ⊔ ¬(¬x)
proof –
from huntington have
\[ x \cup -x = -((-x) \cup -(-x)) \cup -((-x) \cup -(-x)) \cup
\]
(\[ -((-x) \cup -(-x)) \cup -(-x) \cup -(-x) \])
by simp
also from sup-comm have
\[ \ldots = -((-x) \cup -(-x)) \cup -((-x) \cup -(-x)) \cup
\]
\[ -(-x) \cup -(-x) \cup -(-x) \cup -(-x) \]
by simp
also from sup-assoc have
\[ \ldots = -((-x) \cup -(-x)) \cup
\]
\[ -(-x) \cup -(-x) \cup -(-x) \cup -(-x) \]
by simp
also from sup-comm have
\[ \ldots = -((-x) \cup -(-x)) \cup -(-x) \cup -(-x) \cup -(-x) \]
by simp
also from sup-assoc have
\[ \ldots = -((-x) \cup -(-x)) \cup -(-x) \cup -(-x) \cup -(-x) \]
by simp
also from huntington have
\[ \ldots = -x \cup -(\neg x) \]
by simp
finally show ?thesis by simp
qed

lemma dbl-neg: \((\neg x) = x\)
apply (metis huntington huntington-id sup-comm)
done

lemma towards-sup-compl: \(x \cup -x = y \cup -y\)
proof
from huntington have
\[ x \cup -x = -(-x \cup -(\neg y)) \cup -(-x \cup -y) \cup \neg(-x \cup -y) \cup -(-x \cup -y) \]
by simp
also from sup-comm have
\[ \ldots = -(-x \cup -(\neg y)) \cup -(-x \cup -y) \cup \neg(-x \cup -(\neg y)) \cup -(\neg x \cup -(\neg y)) \]
by simp
also from sup-assoc have
\[ \ldots = -(-x \cup -(\neg y)) \cup -(\neg y \cup -(\neg x)) \cup -(\neg y \cup -(\neg x)) \]
by simp
also from sup-comm have
\[
\ldots = (y \lor -x) \lor (-y \lor -x) \lor (-y \lor -x) \lor (-y \lor -x) \\
\text{by simp}
\]
also from sup-assoc have
\[
\ldots = (y \lor -x) \lor (-y \lor -x) \lor (-y \lor -x) \lor (-y \lor -x) \\
\text{by simp}
\]
also from sup-comm have
\[
\ldots = (y \lor -x) \lor (-y \lor -x) \lor (-y \lor -x) \lor (-y \lor -x) \\
\text{by simp}
\]
also from huntington have
\[
y \lor -y = \ldots \text{ by simp}
\]
finally show \( \triangleright \)thesis by simp
qed

\textbf{lemma} sup-compl: \( x \vert -x = T \)
by (simp add: top-eq towards-sup-compl)

\textbf{lemma} towards-inf-compl: \( x \cap -x = y \cap -y \)
by (metis dbl-neg inf-eq sup-comm sup-compl)

\textbf{lemma} inf-compl: \( x \cap -x = \bot \)
by (metis dbl-neg sup-comm bot-eq towards-inf-compl inf-eq)

\textbf{lemma} towards-idem: \( \bot = \bot \lor \bot \)
by (metis dbl-neg inf-eq sup-assoc sup-comm sup-compl)

\textbf{lemma} sup-ident: \( x \lor \bot = x \)
by (metis dbl-neg huntington inf-compl inf-eq sup-assoc sup-comm sup-compl
      towards-idem)

\textbf{lemma} sup-nil: \( x \lor \top = \top \)
by (metis sup-idem sup-assoc sup-comm sup-compl)

\textbf{lemma} inf-nil: \( x \cap \bot = \bot \)
by (metis dbl-neg inf-compl inf-eq sup-nil sup-comm sup-compl)

\textbf{lemma} sup-absorb: \( x \lor x \cap y = x \)
by (metis huntington inf-eq sup-idem sup-assoc sup-comm)

\textbf{lemma} inf-absorb: \( x \cap (x \lor y) = x \)
by (metis dbl-neg inf-eq sup-absorb)
lemma partition: \( x \cap y \cup x \cap -y = x \)
by (metis dbl-neg huntington inf-eq sup-comm)

lemma demorgans1: \(- (x \cap y) = -x \cup -y \)
by (metis dbl-neg inf-eq)

lemma demorgans2: \(- (x \cup y) = -x \cap -y \)
by (metis dbl-neg inf-eq)

lemma inf-comm: \( x \cap y = y \cap x \)
by (metis inf-eq sup-comm)

lemma inf-assoc: \( x \cap (y \cap z) = (x \cap y) \cap (x \cap z) \)
by (metis dbl-neg inf-eq sup-assoc)

lemma inf-sup-distrib1: \( x \cap (y \sqcup z) = (x \cap y) \sqcup (x \cap z) \)
proof –
from partition have
\( x \cap (y \sqcup z) = x \cap (y \cup z) \cap y \cup x \cap (y \cup z) \cap -y \)
also from inf-assoc have
\( \ldots = x \cap ((y \cup z) \cap y) \cup x \cap (y \cup z) \cap -y \) by simp
also from inf-comm have
\( \ldots = x \cap (y \cap (y \cup z)) \cup x \cap (y \cup z) \cap -y \) by simp
also from inf-absorb have
\( \ldots = (x \cap y) \sqcup (x \cap (y \cup z) \cap -y) \) by simp
also from partition have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad ((x \cap (y \cup z) \cap -y \cap z) \cup (x \cap (y \cup z) \cap -y \cap -z)) \) by simp
also from inf-assoc have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad ((x \cap ((y \cup z) \cap -y \cap z)) \cup (x \cap ((y \cup z) \cap -y \cap -z))) \) by simp
also from demorgans2 have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad ((x \cap ((y \cup z) \cap -y \cap z)) \cup (x \cap ((y \cup z) \cap -y \cap -z))) \) by simp
also from inf-compl have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad ((x \cap ((y \cup z) \cap -y \cap z)) \cup (x \cap \bot)) \) by simp
also from inf-nil have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad ((x \cap ((y \cup z) \cap -y \cap z)) \cup \bot) \) by simp
also from sup-idem have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad ((x \cap ((y \cup z) \cap -y \cap z)) \cup \bot) \) by simp
also from sup-ident have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\( \quad (x \cap ((y \cup z) \cap -y \cap -z)) \) by simp
also from inf-comm have
\( \ldots = ((x \cap y \cap z) \sqcup (x \cap y \cap -z)) \sqcup \)
\[(x \cap ((-y \cap z) \cap (y \cup z))) \text{ by simp}\]

also from sup-comm have
\[\ldots = ((x \cap y \cap z) \cup (x \cap y \cap z) \cup (x \cap y \cap -z)) \cup (x \cap ((-y \cap z) \cap (z \cup y))) \text{ by simp}\]

also from inf-assoc have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (-y \cap z))\]
by simp

also from inf-absorb have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (z \cap -y))\]
by simp

also from inf-comm have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (z \cap -y))\]
by simp

also from sup-assoc have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (z \cap -y))\]
by simp

also from sup-comm have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (z \cap -y))\]
by simp

also from inf-assoc have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (z \cap -y))\]
by simp

also from inf-absorb have
\[\ldots = ((x \cap y \cap z) \cup (x \cap (y \cap z)) \cup (x \cap y \cap -z)) \cup (x \cap (z \cap -y))\]
by simp

finally show thesis have \[\ldots = (x \cap y) \cup (x \cap z)\]
by simp

qed

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

proof –
from dbl-neg have
\[x \cup (y \cap z) = -((-(-(-x) \cup (y \cap z))) \text{ by simp}\]

also from inf-eq have
\[\ldots = -((-x \cap (-y \cup -z)) \text{ by simp}\]

also from inf-sup-distrib1 have
\[\ldots = -(-x \cap -y) \cup (-x \cap -z) \text{ by simp}\]

also from demorgans2 have
\[\ldots = -(-x \cap -y) \cap -(-x \cap -z) \text{ by simp}\]

also from demorgans1 have
\[\ldots = (-(-x) \cup -(y)) \cap (-(-x) \cup -(z)) \text{ by simp}\]

also from dbl-neg have
\[\ldots = (x \cup y) \cap (x \cup z) \text{ by simp}\]

finally show thesis by simp

qed

lemma huntington-is-boolean-II:
\[\text{class boolean-algebra-II uminus } (\cap) (\cup) \bot \top\]

apply unfold-locale

apply
apply (metis inf-comm inf-assoc sup-absorb
inf-absorb sup-inf-distrib1
sup-compl inf-compl)+
done

lemma huntington-is-boolean:
class.boolean-algebra minus uminus (⊓) (⊔) (⊏) (⊔) ⊥ ⊤
proof –
interpret boolean-II:
boolean-algebra-II uminus (⊓) ⊥ ⊤
by (fact huntington-is-boolean-II)
show ?thesis by (simp add: boolean-II.boolean-II-is-boolean)
qed
end

3.3 Robbins’ Algebra

context boolean-algebra begin
lemma boolean-is-robbins:
class.robbins-algebra uminus inf sup bot top
apply unfold-locales
apply (metis sup-assoc sup-commute compl-inf double-compl sup-compl-top
inf-compl-bot diff-eq sup-bot-right sup-inf-distrib1)+
done
end

context boolean-algebra-II begin
lemma boolean-II-is-robbins:
class.robbins-algebra uminus inf sup bot top
proof –
interpret boolean:
boolean-algebra minus uminus (⊓) (⊔) ⊥ ⊤
by (fact boolean-II-is-boolean)
show ?thesis by (simp add: boolean.boolean-is-robbins)
qed
end

context huntington-algebra begin
lemma huntington-is-robbins:
class.robbins-algebra uminus inf sup bot top
proof –
interpret boolean:
boolean-algebra minus uminus (⊓) (⊔) (⊔) ⊥ ⊤
by (fact huntington-is-boolean)
show ?thesis by (simp add: boolean.boolean-is-robbins)
qed
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 –
  have 1 = Suc(0) by arith
  hence 1 × x = Suc(0) × x bymetis
  also have . . . = x by simp
  finally show thesis by simp
qed

lemma two: 2 × x = x ⊔ x
proof –
  have 2 = Suc(Suc(0)) by arith
  hence 2 × x = Suc(Suc(0)) × x bymetis
  also have . . . = x ⊔ x by simp
  finally show thesis by simp
qed

lemma three: 3 × x = x ⊔ x ⊔ x
proof –
  have 3 = Suc(Suc(Suc(0))) by arith
  hence 3 × x = Suc(Suc(Suc(0))) × x bymetis
  also have . . . = x ⊔ x ⊔ x by simp
  finally show thesis by simp
qed

lemma four: 4 × x = x ⊔ x ⊔ x ⊔ x
proof –
  have 4 = Suc(Suc(Suc(Suc(0)))) by arith
  hence 4 × x = Suc(Suc(Suc(Suc(0)))) × x bymetis
  also have . . . = x ⊔ x ⊔ x ⊔ x by simp
  finally show thesis by simp

10
qed

**Lemma five:** \(5 \times x = x \sqcup x \sqcup x \sqcup x \sqcup x\)

**Proof** –

- **have** \(5 = \text{Suc(\text{Suc(\text{Suc(\text{Suc(0)}}))})}\) by arith
- **hence** \(5 \times x = \text{Suc(\text{Suc(\text{Suc(\text{Suc(\text{Suc(0)}}))})})} \times x\) by metis
- **also have** \(\ldots = x \sqcup x \sqcup x \sqcup x \sqcup x\) by simp
- **finally show** \(\text{?thesis}\) by simp

qed

**Lemma six:** \(6 \times x = x \sqcup x \sqcup x \sqcup x \sqcup x \sqcup x\)

**Proof** –

- **have** \(6 = \text{Suc(\text{Suc(\text{Suc(\text{Suc(\text{Suc(0)}}))})})}\) by arith
- **hence** \(6 \times x = \text{Suc(\text{Suc(\text{Suc(\text{Suc(\text{Suc(\text{Suc(0)}}))})})})} \times x\) by metis
- **also have** \(\ldots = x \sqcup x \sqcup x \sqcup x \sqcup x \sqcup x\) by simp
- **finally show** \(\text{?thesis}\) by simp

qed

**Lemma copyp-distrib:** \(k \otimes (x \sqcup y) = (k \otimes x) \sqcup (k \otimes y)\)

**Proof** (induct \(k\))

- **case** \(0\) **show** \(\text{?case}\) by simp
- **case** \(\text{Suc}\) **thus** \(\text{?case}\) by (simp, metis sup-assoc sup-comm)

qed

**Corollary copyp-distrib:** \(k \times (x \sqcup y) = (k \times x) \sqcup (k \times y)\)

**by** (induct \(k\), (simp add: sup-assoc sup-comm copyp-distrib)+)

**Lemma copyp-arith:** \((k + l + 1) \otimes x = (k \otimes x) \sqcup (l \otimes x)\)

**Proof** (induct \(l\))

- **case** \(0\) **have** \(k + 0 + 1 = \text{Suc(k)}\) by arith
  - **thus** \(\text{?case}\) by simp
- **case** \((\text{Suc(l)}\) **note** ind-hyp = this
  - **have** \(k + \text{Suc(l)} + 1 = \text{Suc(k + l + 1)}\) by arith+
  - **hence** \((k + \text{Suc(l)} + 1) \otimes x = (k + l + 1) \otimes x \sqcup x\) by (simp add: ind-hyp)
  - **also from** ind-hyp **have** \(\ldots = (k \otimes x) \sqcup (l \otimes x) \sqcup x\) by simp
  - **also note** sup-assoc
  - **finally show** \(\text{?case}\) by simp

qed

**Lemma copy-arith:**

- **assumes** \(k \neq 0\) and \(l \neq 0\)
- **shows** \((k + l) \times x = (k \times x) \sqcup (l \times x)\)

**Using** assms

**Proof** –

- **from** assms **have** \(\exists k'. \text{Suc(k')} = k\)
  - **and** \(\exists l'. \text{Suc(l')} = l\) by arith+
from this obtain $k' l'$ where $A: \text{Suc}(k') = k$ and $B: \text{Suc}(l') = l$ by fast+
from this have $A1: k \times x = k' \otimes x$ and $B1: l \times x = l' \otimes x$ by fastforce+
from $A \ B$ have $k + l = \text{Suc}(k' + l' + 1)$ by arith
hence $(k + l) \times x = (k' + l' + 1) \otimes x$ by simp
also from copyyp-arith have
\[ \ldots = k' \otimes x \sqcup l' \otimes x \text{ by fast} \]
also from $A1 \ B1$ have
\[ \ldots = k \times x \sqcup l \times x \text{ by fastforce} \]
finally show ?thesis by simp
qed

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
$\alpha = -(-t \sqcup t \sqcup t) \sqcup t$
definition secret-object3 :: 'a ($\beta$) where
$\beta = t \sqcup t$
definition secret-object4 :: 'a ($\delta$) where
$\delta = \beta \sqcup (-($$\alpha \sqcup -$$\delta) \sqcup -($$\alpha \sqcup -$$\delta))$
definition secret-object5 :: 'a ($\gamma$) where
$\gamma = \delta \sqcup -($$\delta \sqcup -$$\delta)$
definition winker-object :: 'a ($\rho$) where
$\rho = \gamma \sqcup \gamma \sqcup \gamma$
definition fake-bot :: 'a ($\bot \bot$) where
$\bot \bot = -(\rho \sqcup -\rho)$

lemma robbins2: $y = -(-(x \sqcup y) \sqcup -(x \sqcup y))$
by (metis robbins sup-comm)
lemma mann0: $-(x \sqcup y) = -(-(x \sqcup y) \sqcup -(x \sqcup y) \sqcup y)$
by (metis robbins sup-comm sup-assoc)


lemma mann1: \(-x \sqcup y\) = \(-(\neg x \sqcup y) \cup y\) 
by (metis robbins sup-comm sup-assoc)

lemma mann2: \(y = \neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y\) 
by (metis mann1 robbins sup-comm sup-assoc)

lemma mann3: \(z = \neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y) \cup\neg(\neg x \sqcup y) \cup z) \cup (\neg y \sqcup z)\)
proof
  let \(\forall w = \neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y\) \cup \neg(\neg x \sqcup y) \cup (\neg y \sqcup z)\)
  from robbins[where \(x=z\) and \(y=?w\)] sup-comm mann2
  have \(z = \neg(\neg y \sqcup z) \cup (\neg w \sqcup z)\) by metis
  thus \(?thesis by (metis sup-comm)\)
qed

lemma mann4: \((y \sqcup z) = (\neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y) \cup\neg(\neg x \sqcup y) \cup (y \sqcup z) \cup z) \cup (\neg y \sqcup z)\)
proof
  from robbins2[where \(x=\neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y) \cup\neg(\neg x \sqcup y) \cup (y \sqcup z) \cup z) \cup (\neg y \sqcup z)\)
  and \(y=-\(y \sqcup z)\]
  mann3[where \(x=x\) and \(y=y\) and \(z=z\)]
  have \((y \sqcup z) = \neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y) \cup\neg(\neg x \sqcup y) \cup (y \sqcup z) \cup z\)
  by metis
  with sup-comm sup-assoc show \(?thesis by metis\)
qed

lemma mann5: \(u = (\neg(\neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y) \cup\neg(\neg x \sqcup y) \cup (y \sqcup z) \cup z) \cup (\neg y \sqcup z)\)
using robbins2[where \(x=\neg(\neg x \sqcup y) \cup x \sqcup y \sqcup y) \cup\neg(\neg x \sqcup y) \cup (y \sqcup z) \cup z) \cup (\neg y \sqcup z)\)
  and \(y=u\]
  mann4[where \(x=x\) and \(y=y\) and \(z=z\)]
  sup-comm
  by metis

lemma mann6: \((\neg 3 \times x \cup x) = \neg(\neg(\neg 3 \times x \cup x) \cup 3 \times x) \cup\neg(\neg 3 \times x \cup x) \cup (5 \times x)\)
proof
  have \(3+2=(5::nat)\) and \(3\neq(6::nat)\) and \(2\neq(6::nat)\) by arith+
  with copy-arith have \(\triangledown: 3 \times x \cup 2 \times x = 5 \times x\) by metis
  let \(?p = \neg(\neg 3 \times x \cup x)\)
  { fix \(q\)
    from sup-comm have \((q \cup 5 \times x) = \neg(5 \times x \cup q)\) by metis
    also from \(\triangledown\) mann6[where \(x=3 \times x\) and \(y=q\) \cup 2 \times x] sup-assoc sup-comm
  }
have

\[ \ldots = \neg(-(-((3 \times x \cup (q \cup 2 \times x)) \cup -3 \times x \cup (q \cup 2 \times x)) \cup (q \cup 2 \times x)) \] by metis

also from sup-assoc have

\[ \ldots = \neg(-(-((3 \times x \cup q) \cup 2 \times x) \cup -3 \times x \cup (q \cup 2 \times x)) \cup (q \cup 2 \times x)) \] by metis

also from sup-comm have

\[ \ldots = \neg(-((q \cup 3 \times x) \cup 2 \times x) \cup -3 \times x \cup (q \cup 2 \times x)) \cup (q \cup 2 \times x)) \] by metis

also from sup-assoc have

\[ \ldots = \neg(-(q \cup (3 \times x \cup 2 \times x)) \cup -3 \times x \cup (q \cup 2 \times x)) \cup (q \cup 2 \times x)) \] by metis

also from \( \sqcup \) have

\[ \ldots = \neg(-(q \cup 5 \times x) \cup -3 \times x \cup (q \cup 2 \times x)) \cup (q \cup 2 \times x)) \] by metis

also from sup-assoc have

\[ \ldots = \neg(-(q \cup 5 \times x) \cup -(3 \times x \cup q) \cup 2 \times x) \cup (q \cup 2 \times x)) \] by metis

also from sup-comm have

\[ \ldots = \neg(-(q \cup 5 \times x) \cup (q \cup -3 \times x) \cup 2 \times x) \cup (2 \times x \cup q)) \] by metis

also from sup-assoc have

\[ \ldots = \neg(-(q \cup 5 \times x) \cup q \cup -3 \times x \cup 2 \times x) \cup 2 \times x \cup q) \] by metis

finally have

\[ -(q \cup 5 \times x) = \neg(-(q \cup 5 \times x) \cup q \cup -3 \times x \cup 2 \times x) \cup 2 \times x \cup q) \] by simp

\( \) hence \( \blacklozenge: \)

\[ -(?p \cup 5 \times x) = \neg(-(?p \cup 5 \times x) \cup ?p \cup -3 \times x \cup 2 \times x) \cup 2 \times x \cup ?p) \]

by simp

from mann5[where \( x=3 \times x \) and \( y=x \) and \( z=2 \times x \) and \( u=\neg ?p \)]

\( \) sup-assoc three[where \( x=x \)] five[where \( x=x \)] have

\( ?p = \)

\[ \neg(-(-(?p \cup 5 \times x) \cup ?p \cup -(x \cup 2 \times x) \cup 2 \times x) \cup 2 \times x \cup ?p) \cup \]

\[ -(x \cup 2 \times x) \cup ?p) \] by metis

also from sup-comm have

\[ \ldots = \neg(-(-(?p \cup 5 \times x) \cup ?p \cup -(2 \times x \cup x) \cup 2 \times x) \cup 2 \times x \cup ?p) \cup \]

\[ -(2 \times x \cup x) \cup ?p) \] by metis

also from two[where \( x=x \)] three[where \( x=x \)] have

\[ \ldots = \neg(-(?p \cup 5 \times x) \cup ?p \cup -3 \times x \cup 2 \times x) \cup 2 \times x \cup ?p) \cup \]

\[ -(-3 \times x \cup ?p) \] by metis

also from \( \blacklozenge \) have \( \ldots = \neg(-(?p \cup 5 \times x) \cup -(3 \times x \cup ?p)) \) by simp

also from sup-comm have \( \ldots = \neg(-(?p \cup 5 \times x) \cup -(?p \cup -3 \times x)) \) by simp

also from sup-comm have \( \ldots = \neg(-(?p \cup -3 \times x) \cup -(?p \cup 5 \times x)) \) by simp

finally show \( ?\)thesis.

qed

lemma mann7:  
\[ -3 \times x = \neg(-(3 \times x \cup x) \cup 5 \times x) \]

proof –

let \( ?p = \neg(-3 \times x \cup x) \)

let \( ?q = ?p \cup -3 \times x \)

let \( ?r = -(?p \cup 5 \times x) \)

from robbins2[where \( x=?q \)
and $y = ?r$

\[ \text{mann9[where } x = x] \]
\begin{align*}
\text{have } \& \text{?r } &= \; (\& \text{?p} \sqcup \; (\& \text{?q} \sqcup \; ?r)) \text{ by simp} \\
\text{also from sup-comm have } \ldots &= \; (\& \text{?q} \sqcup \; ?r) \text{ by simp} \\
\text{also from sup-comm have } \ldots &= \; (\& \text{?r} \sqcup \; ?q) \text{ by simp} \\
\text{finally have } \& \text{?r } &= \; (\& \text{?r} \sqcup \; ?q) . \\
\text{from mann3[where } x = 3 \times x \text{ and } y = x \text{ and } z = 3 \times x] \\
\text{sup-comm have } \\
\ldots &= \; (\& \text{?r} \sqcup \; ?q) \text{ by simp} \\
\text{finally have } \& \text{?thesis} & \; \text{by simp} \\
\text{qed}
\end{align*}

\begin{enumerate}
\item \textbf{lemma mann8:}
\begin{align*}
&\ldots = \; (-(-3 \times x \sqcup x) \sqcup 2 \times x) \sqcup (-(-3 \times x \sqcup x) \sqcup -3 \times x \sqcup 2 \times x) \sqcup -3 \times x) \\
\text{(is } \&lhs = \; ?lhs) \\
\text{proof } \\
\text{let } \&p &= \; (-3 \times x \sqcup x) \\
\text{let } \&q &= \; ?p \sqcup 2 \times x \\
\text{let } \&r &= \; 3 \times x \\
\text{have } 3 + 2 &= (5 : \text{nat}) \text{ and } 3 \not\in (\&lhs : \text{nat}) \text{ and } 2 \not\in (\&lhs : \text{nat}) \text{ by arith+} \\
\text{with copy-arith have } \& \text{?thesis by simp} \\
\text{finally show } \& \text{?thesis by simp} \\
\text{qed}
\end{align*}
\item \textbf{lemma mann9:}
\begin{align*}
&\ldots = \; (-(-3 \times x \sqcup x) \sqcup -3 \times x \sqcup 2 \times x) \sqcup -3 \times x) \\
\text{proof } \\
\text{let } \&p &= \; (-3 \times x \sqcup x) \\
\text{let } \&q &= \; ?p \sqcup 4 \times x \\
\text{have } 4 + 1 &= (5 : \text{nat}) \text{ and } 1 \not\in (\&lhs : \text{nat}) \text{ and } 4 \not\in (\&lhs : \text{nat}) \text{ by arith+} \\
\text{with copy-arith one have } \& \text{?thesis by simp} \\
\text{finally show } \& \text{?thesis by simp} \\
\text{qed}
\end{align*}
\end{enumerate}
with sup-assoc robbins2[where y=x and x=?q]
have x = −(−(?q ⊔ x) ⊔ −(?p ⊔ 5×x)) by metis
with mann7 have x = −(−(?q ⊔ x) ⊔ −3×x) by metis
moreover
have 3+1=(4::nat) and 1≠(0::nat) and 3≠(0::nat) by arith+
with copy-arith one have ♠: 3×x ⊔ y = 4×x by metis
ultimately show ?thesis by simp
qed

lemma mann10: y = −(−(−3×x ⊔ x) ⊔ −3×x ⊔ y) ⊔ −(x ⊔ y))
using robbins2[where x=−(−3×x ⊔ x) ⊔ −3×x and y=y]
  mann9[where x=x]
  sup-comm
by metis

theorem mann: 2×x = −(−3×x ⊔ x) ⊔ 2×x
using mann10[where x=x and y=2×x]
  mann8[where x=x]
  two[where x=x] three[where x=x] sup-comm
by metis

corollary winkerr: α ⊔ β = β
using mann secret-object2-def secret-object3-def two three
by metis

corollary winker: β ⊔ α = β
by (metis winkerr sup-comm)

corollary multi-winkerr: β ⊔ k ⊗ α = β
by (induct k, (simp add: winkerr sup-comm sup-assoc)+)

corollary multi-winker: β ⊔ k × α = β
by (induct k, (simp add: multi-winkerr winker sup-comm sup-assoc)+)

lemma less-eq-introp:
−(x ⊔ −(y ⊔ z)) = −(x ⊔ y ⊔ −z) ⇒ y ⊑ x
by (metis robbins sup-assoc less-eq-def
sup-comm[where x=x and y=y])

corollary less-eq-intro:
−(x ⊔ −(y ⊔ z)) = −(x ⊔ y ⊔ −z) ⇒ x ⊔ y = x
by (metis less-eq-introp less-eq-def sup-comm)

lemma eq-intro:
−(x ⊔ −(y ⊔ z)) = −(y ⊔ −(x ⊔ z)) ⇒ x = y
by (metis robbins sup-assoc sup-comm)

lemma copyp0:
  assumes $- (x \sqcup - y) = z$
  shows $- (x \sqcup - (y \sqcup k \otimes (x \sqcup z))) = z$
using assms
proof (induct k)
  case 0 show ?case by (simp, metis assms robbins sup-assoc sup-comm)
  case Suc note ind-hyp = this show ?case by (simp, metis ind-hyp robbins sup-assoc sup-comm)
qed

lemma copyp1:
  assumes $- (- (x \sqcup - y) \sqcup - y) = x$
  shows $- (y \sqcup k \otimes (x \sqcup - (x \sqcup - y))) = - y$
using assms
proof
  let ?z = $- (x \sqcup - y)$
  let ?ky = $y \sqcup k \otimes (x \sqcup ?z)$
  have $- (x \sqcup - ?ky) = ?z$ by (simp add: copyp0)
  hence $- (?ky \sqcup - (- y \sqcup ?z)) = ?z$ by (metis assms sup-comm)
  also have $- (?z \sqcup - ?ky) = x$ by (metis assms copyp0 sup-comm)
  hence $?z = - (- y \sqcup - (?ky \sqcup ?z))$ by (metis sup-comm)
  finally show ?thesis by (metis eq-intro)
qed

corollary copyp2:
  assumes $- (x \sqcup y) = - y$
  shows $- (y \sqcup k \otimes (x \sqcup - (x \sqcup - y))) = - y$
  by (metis assms robbins sup-comm copyp1)

lemma two-threep:
  assumes $- (2 \times x \sqcup y) = - y$
  and $- (3 \times x \sqcup y) = - y$
  shows $2 \times x \sqcup y = 3 \times x \sqcup y$
using assms
proof
  from assms two three have
    A: $- (x \sqcup x \sqcup y) = - y$ and 
    B: $- (x \sqcup x \sqcup x \sqcup y) = - y$ by simp+
  with sup-assoc
    copyp2[where x=x and y=x \sqcup x \sqcup y and k=0]
  have $- (x \sqcup x \sqcup y \sqcup x \sqcup - (x \sqcup - y)) = - y$ by simp
  moreover
  from sup-comm sup-assoc A B
    copyp2[where x=x \sqcup x and y=y and k=0]
  have $- (y \sqcup x \sqcup x \sqcup - (x \sqcup x \sqcup - y)) = - y$ by fastforce

with sup-comm sup-assoc
have $- (x \sqcup x \sqcup y \sqcup -(x \sqcup (x \sqcup -y))) = -y$ by metis
ultimately have
$-(x \sqcup x \sqcup y \sqcup -(x \sqcup -y))) = -(x \sqcup x \sqcup y \sqcup x \sqcup -(x \sqcup -y))$ by simp
with less-eq-intro have $x \sqcup x \sqcup y = x \sqcup x \sqcup y \sqcup x$ by metis
with sup-comm sup-assoc two three show thesis by metis
qed

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 $y \sqcup ?z2 = y \sqcup ?z3$)
using assms
proof
assume $-(x \sqcup y) = -y$
with copyp2 [where $k = \text{Suc}(0)$]
  copyp2 [where $k = \text{Suc}(\text{Suc}(0))$]
two three
have $- (y \sqcup ?z2) = -y$ and $- (y \sqcup ?z3) = -y$ by simp+
with two-three sup-comm show thesis by metis
next
assume $-(x \sqcup -y) \sqcup -y = x$
with copyp1 [where $k = \text{Suc}(0)$]
  copyp1 [where $k = \text{Suc}(\text{Suc}(0))$]
two three
have $- (y \sqcup ?z2) = -y$ and $- (y \sqcup ?z3) = -y$ by simp+
with two-three sup-comm show thesis by metis
qed

lemma sup-idem: $\rho \sqcup \rho = \rho$
proof
  from winkerr two
    copyp2 [where $x = \alpha$ and $y = \beta$ and $k = \text{Suc}(0)$] have
  $-\beta = -(\beta \sqcup 2 \times (\alpha \sqcup -(\alpha \sqcup -\beta)))$ by simp
also from copy-distrib sup-assoc have
... $= -(\beta \sqcup 2 \times \alpha \sqcup 2 \times (-(\alpha \sqcup -\beta)))$ by simp
also from sup-assoc secret-object4-def two
  multi-winker [where $k = 2$] have
... $= -\delta$ by metis
finally have $-\beta = -\delta$ by simp
with secret-object4-def sup-assoc three have
$\delta \sqcup -(\alpha \sqcup -\delta) = \beta \sqcup 3 \times (-(\alpha \sqcup -\beta))$ by simp
also from copy-distrib [where $k = 3$]
  multi-winker [where $k = 3$]
sup-assoc have
... $= \beta \sqcup 3 \times (\alpha \sqcup -(\alpha \sqcup -\beta))$ by metis
also from wink sup-comm two-three [where $x = \alpha$ and $y = \beta$] have
... $= \beta \sqcup 2 \times (\alpha \sqcup -(\alpha \sqcup -\beta))$ by fastforce
also from copy-distrib [where $k = 2$]
multi-winker[where \(k=2\)]

sup-assoc two secret-object4-def have

\[\ldots = \delta \text{ by metis}\]

finally have \(\triangledown: \delta \uplus -(\alpha \uplus -\delta) = \delta\) by simp

from secret-object4-def winker sup-assoc have

\(\alpha \uplus \delta = \delta\) by metis

hence \(\delta \uplus \alpha = \delta\) by (metis sup-comm)

hence \(-(\delta \uplus -\delta) \uplus -\delta) = -(\(\delta \uplus (\alpha \uplus -\delta)) \uplus -\delta)\) by (metis sup-assoc)

also from \(\triangledown\) have

\(\ldots = -(\(\delta \uplus (\alpha \uplus -\delta)) \uplus -(\delta \uplus -(\alpha \uplus -\delta)))\) by metis

also from robbins have

\(\ldots = \delta\) by metis

finally have \(-(\(\delta \uplus -\delta) \uplus -\delta) = \delta\) by simp

with two-three[where \(x=\delta\) and \(y=\delta\)]

secret-object5-def sup-comm

have \(3 \times \gamma \uplus \delta = 2 \times \gamma \uplus \delta\) by fastforce

with secret-object5-def sup-assoc sup-comm have

\(3 \times \gamma \uplus \gamma = 2 \times \gamma \uplus \gamma\) by fastforce

with two three four five six have

\(6 \times \gamma = 3 \times \gamma\) by simp

moreover have \(3 + 3 = (6::\text{nat})\) and \(3 \neq (0::\text{nat})\) by arith+

moreover note copy-arith[where \(k=3\) and \(l=3\) and \(x=\gamma\)]

winker-object-def three

ultimately show \(\text{thesis}\) by simp

qed

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

proof –

have \(I: g = -(\neg g \uplus \bot \bot)\)

by (metis fake-bot-def inf-eq robbins sup-comm sup-idem)

\{ fix \(x\) have \(x = -(\neg (x \uplus \neg g \uplus \bot \bot) \uplus -(x \uplus g))\)

by (metis \(I\) robbins sup-assoc) \}

note \(II = \text{this}\)

have \(III: -g = -(\neg (\neg g \uplus \neg g \uplus \neg g) \uplus g)\)

by (metis robbins[where \(x=-g\) and \(y=g \uplus \neg g\])

\(I\) sup-comm fake-bot-def)

hence \(g = -(\neg (\neg g \uplus \neg g \uplus \neg g) \uplus \neg g)\)

by (metis robbins[where \(x=g\) and \(y=g \uplus \neg g \uplus \neg g\])

sup-comm[where \(x=g\) and \(y=-\neg (\neg g \uplus \neg g \uplus \neg g)\])

sup-assoc sup-idem)

hence \(\neg (\neg g \uplus \neg g \uplus \neg g) = \bot \bot\)

by (metis robbins[where \(x=-\neg g \uplus \neg g\) and \(y=g\])

III sup-comm fake-bot-def)

hence \(-g = -(\neg g \uplus \bot \bot)\)

by (metis \(III\) sup-comm)

qed
hence $\varrho = -(-\varrho \sqcup \bot \sqcup -\varrho)$
by (metis II sup-idem sup-comm sup-assoc)
moreover have $\varrho \sqcup \bot \bot = -((-\varrho \sqcup \bot \bot) \sqcup -(-\varrho))$
by (metis robbins[where $x=\varrho \sqcup \bot \bot$ and $y=\varrho$]
    sup-comm[where $y=\varrho$]
    sup-assoc sup-idem)
ultimately have $\varrho = \varrho \sqcup \bot \bot$
by auto
hence $x \sqcup \bot \bot = -((-x \sqcup \varrho) \sqcup -(x \sqcup \bot \bot \sqcup -\varrho))$
by (metis robbins[where $x=x \sqcup \bot \bot$ and $y=\varrho$]
    sup-comm[where $x=\bot \bot$ and $y=\varrho$]
    sup-assoc)
thus $\vartheta$thesis by (metis sup-assoc sup-comm II)
qed

lemma dbl-neg: $-(-x) = x$
proof
  { fix $x$ have $\bot \bot = -(-x \sqcup -(x))$
    by (metis robbins sup-comm sup-ident)
  } note I = this

  { fix $x$ have $-x = -((-x \sqcup -(x))))$
    by (metis I robbins sup-comm sup-ident)
  } note II = this

  { fix $x$ have $-(-x) = -((-x \sqcup -(x))))$
    by (metis I II robbins sup-assoc sup-comm sup-ident)
  } note III = this

  show $\vartheta$thesis by (metis II III robbins)
qed

theorem robbins-is-huntington:
class.huntington-algebra uminus ($\sqcap$) ($\sqcup$) $\bot$ $\top$
apply unfold-locales
apply (metis dbl-neg robbins sup-comm)
done

theorem robbins-is-boolean-II:
class.boolean-algebra-II uminus ($\sqcap$) ($\sqcup$) $\bot$ $\top$
proof
  interpret huntington:
  class.huntington-algebra uminus ($\sqcap$) ($\sqcup$) $\bot$ $\top$
by (fact robbins-is-huntington)
  show $\vartheta$thesis by (simp add: huntington.huntington-is-boolean-II)
qed
**theorem** robbins-is-boolean:

\[
\text{class.\ boolean-algebra minus uminus (\(\cap\) (\(\subseteq\) (\(\sqsubseteq\) (\(\sqsubseteq\) (\(\sqcup\) (\(\bot\) \(\top\))

**proof**

- **interpret** huntington:

  \[
  \text{huntington-algebra uminus (\(\cap\) (\(\cup\)) (\(\bot\) \(\top\))
  \]

  **by** (fact robbins-is-huntington)

- **show** \(?thesis\) **by** (simp add: huntington.huntington-is-boolean)

  **qed**

**end**

**no-notation** secret-object1 (\(\iota\))

- **and** secret-object2 (\(\alpha\))

- **and** secret-object3 (\(\beta\))

- **and** secret-object4 (\(\delta\))

- **and** secret-object5 (\(\gamma\))

- **and** winker-object (\(\rho\))

- **and** less-eq (infix \(\subseteq\) 50)

- **and** less (infix \(\subset\) 50)

- **and** inf (infixl \(\sqsubseteq\) 70)

- **and** sup (infixl \(\sqcup\) 65)

- **and** top (\(\top\))

- **and** bot (\(\bot\))

- **and** copyp (infix \(\otimes\) 80)

- **and** copy (infix \(\times\) 85)

**notation**

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

**end**