A Hierarchy of Algebras for Boolean Subsets

Walter Guttmann and Bernhard Möller

March 17, 2025

Abstract

We present a collection of axiom systems for the construction of Boolean subalgebras of larger overall algebras. The subalgebras are defined as the range of a complement-like operation on a semilattice. This technique has been used, for example, with the antidomain operation, dynamic negation and Stone algebras. We present a common ground for these constructions based on a new equational axiomatisation of Boolean algebras.

Contents

1 Overview

A Boolean algebra often arises as a subalgebra of some overall algebra. To avoid introducing a separate type for the subalgebra, the overall algebra can be enriched with a special operation leading into the intended subalgebra and axioms to guarantee that the range of this operation has a Boolean structure. Examples for this are the antidomain operation in idempotent (left) semirings [6, 7, 8], dynamic negation [17], the operation yielding tests in [13, 16], and the pseudocomplement operation in Stone algebras [9, 12, 14]. The present development looks at a common ground pattern.

In Sections 2 and 3 we relate various axiomatisations of Boolean algebras from the literature and present a new equational one tailored to our needs. Section 4 adapts this for the construction of Boolean subalgebras of larger overall algebras. In Section 5 we add successively stronger assumptions to the overall algebra. Sections 6, 7 and 8 show how Stone algebras, domain semirings and antidomain semirings fit into this hierarchy.

This Isabelle/HOL theory formally verifies results in [15]. See that paper for further details and related work. Some proofs in this theory have been translated from proofs found by Prover9 [21] using a program we wrote.

theory *Subset-Boolean-Algebras*

imports *Stone-Algebras.P-Algebras*

begin

2 Boolean Algebras

We show that Isabelle/HOL's *boolean-algebra* class is equivalent to Huntington's axioms [18]. See [24] for related results.

2.1 Huntington's Axioms

Definition 1

class *huntington* = *sup* + *uminus* +
  assumes *associative*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
  assumes *commutative*: $x \sqcup y = y \sqcup x$
  assumes *huntington*: $x = -(-x \sqcup y) \sqcup -(-x \sqcup -y)$
begin

lemma *top-unique*:
  $x \sqcup -x = y \sqcup -y$
⟨*proof*⟩

end

2.2 Equivalence to *boolean-algebra* Class

Definition 2

class *extended* = *sup* + *inf* + *minus* + *uminus* + *bot* + *top* + *ord* +
  assumes *top-def*: $top = (THE\ x\ .\ \forall\, y\ .\ x = y \sqcup -y)$
  assumes *bot-def*: $bot = -(THE\ x\ .\ \forall\, y\ .\ x = y \sqcup -y)$
  assumes *inf-def*: $x \sqcap y = -(-x \sqcup -y)$
  assumes *minus-def*: $x - y = -(-x \sqcup y)$
  assumes *less-eq-def*: $x \leq y \longleftrightarrow x \sqcup y = y$
  assumes *less-def*: $x < y \longleftrightarrow x \sqcup y = y \wedge \neg\ (y \sqcup x = x)$

class *huntington-extended* = *huntington* + *extended*
begin

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

lemma *bot-char*:
  $bot = -top$
  ⟨*proof*⟩

subclass *boolean-algebra*
⟨*proof*⟩

end

context *boolean-algebra*
begin

sublocale *ba-he*: *huntington-extended*
⟨*proof*⟩

end

2.3 Stone Algebras

We relate Stone algebras to Boolean algebras.

class *stone-algebra-extended* = *stone-algebra* + *minus* +
  assumes *stone-minus-def*[*simp*]: $x - y = x \sqcap -y$

class *regular-stone-algebra* = *stone-algebra-extended* +
  assumes *double-complement*[*simp*]: $--x = x$
begin

subclass *boolean-algebra*
⟨*proof*⟩

end

context *boolean-algebra*
begin

sublocale *ba-rsa*: *regular-stone-algebra*
⟨*proof*⟩

end

3 Alternative Axiomatisations of Boolean Algebras

We consider four axiomatisations of Boolean algebras based only on join and complement. The first three are from the literature and the fourth, a version using equational axioms, is new. The motivation for Byrne's and the new axiomatisation is that the axioms are easier to understand than Huntington's third axiom. We also include Meredith's axiomatisation.

3.1 Lee Byrne's Formulation A

The following axiomatisation is from [2, Formulation A]; see also [10].

Theorem 3

class *boolean-algebra-1* = *sup* + *uminus* +
  assumes *ba1-associative*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
  assumes *ba1-commutative*: $x \sqcup y = y \sqcup x$
  assumes *ba1-complement*: $x \sqcup -y = z \sqcup -z \longleftrightarrow x \sqcup y = x$
begin

subclass *huntington*
⟨*proof*⟩

end

context *huntington*
begin

sublocale *h-ba1*: *boolean-algebra-1*
⟨*proof*⟩

end

3.2 Lee Byrne's Formulation B

The following axiomatisation is from [2, Formulation B].

Theorem 4

class *boolean-algebra-2* = *sup* + *uminus* +
assumes *ba2-associative-commutative*: $(x \sqcup y) \sqcup z = (y \sqcup z) \sqcup x$
assumes *ba2-complement*: $x \sqcup -y = z \sqcup -z \longleftrightarrow x \sqcup y = x$
begin

subclass *boolean-algebra-1*
⟨*proof*⟩

end

context *boolean-algebra-1*
begin

sublocale *ba1-ba2*: *boolean-algebra-2*
⟨*proof*⟩

end

3.3 Meredith's Equational Axioms

The following axiomatisation is from [22, page 221 (1) {A,N}].

class *boolean-algebra-mp* = *sup* + *uminus* +
assumes *ba-mp-1*: $-(-x \sqcup y) \sqcup x = x$
assumes *ba-mp-2*: $-(-x \sqcup y) \sqcup (z \sqcup y) = y \sqcup (z \sqcup x)$
begin

subclass *huntington*
⟨*proof*⟩

end

context *huntington*
begin

sublocale *mp-h*: *boolean-algebra-mp*

⟨*proof*⟩

end

3.4 An Equational Axiomatisation based on Semilattices

The following version is an equational axiomatisation based on semilattices. We add the double complement rule and that *top* is unique. The final axiom *ba3-export* encodes the logical statement $P \vee Q = P \vee (\neg P \wedge Q)$. Its dual appears in [1].

Theorem 5

class *boolean-algebra-3* = *sup* + *uminus* +
assumes *ba3-associative*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
assumes *ba3-commutative*: $x \sqcup y = y \sqcup x$
assumes *ba3-idempotent*[*simp*]: $x \sqcup x = x$
assumes *ba3-double-complement*[*simp*]: $--x = x$
assumes *ba3-top-unique*: $x \sqcup -x = y \sqcup -y$
assumes *ba3-export*: $x \sqcup -(x \sqcup y) = x \sqcup -y$
begin

subclass *huntington*
⟨*proof*⟩

end

context *huntington*
begin

sublocale *h-ba3*: *boolean-algebra-3*
⟨*proof*⟩

end

4 Subset Boolean Algebras

We apply Huntington's axioms to the range of a unary operation, which serves as complement on the range. This gives a Boolean algebra structure on the range without imposing any further constraints on the set. The obtained structure is used as a reference in the subsequent development and to inherit the results proved here. This is taken from [13, 16] and follows the development of Boolean algebras in [20].

Definition 6

class *subset-boolean-algebra* = *sup* + *uminus* +
assumes *sub-associative*: $-x \sqcup (-y \sqcup -z) = (-x \sqcup -y) \sqcup -z$
assumes *sub-commutative*: $-x \sqcup -y = -y \sqcup -x$
assumes *sub-complement*: $-x = -(--x \sqcup -y) \sqcup -(--x \sqcup --y)$

assumes *sub-sup-closed*: $-x \sqcup -y = --(-x \sqcup -y)$
begin

uniqueness of *top*, resulting in the lemma *top-def* to replace the assumption *sub-top-def*

lemma *top-unique*:
$-x \sqcup --x = -y \sqcup --y$
⟨*proof*⟩

consequences for join and complement

lemma *double-negation*[*simp*]:
$---x = -x$
⟨*proof*⟩

lemma *complement-1*:
$--x = -(-x \sqcup -y) \sqcup -(-x \sqcup --y)$
⟨*proof*⟩

lemma *sup-right-zero-var*:
$-x \sqcup (-y \sqcup --y) = -z \sqcup --z$
⟨*proof*⟩

lemma *sup-right-unit-idempotent*:
$-x \sqcup -x = -x \sqcup -(-y \sqcup --y)$
⟨*proof*⟩

lemma *sup-idempotent*[*simp*]:
$-x \sqcup -x = -x$
⟨*proof*⟩

lemma *complement-2*:
$-x = -(-(-x \sqcup -y) \sqcup -(-x \sqcup --y))$
⟨*proof*⟩

lemma *sup-eq-cases*:
$-x \sqcup -y = -x \sqcup -z \Longrightarrow --x \sqcup -y = --x \sqcup -z \Longrightarrow -y = -z$
⟨*proof*⟩

lemma *sup-eq-cases-2*:
$-y \sqcup -x = -z \sqcup -x \Longrightarrow -y \sqcup --x = -z \sqcup --x \Longrightarrow -y = -z$
⟨*proof*⟩

end

Definition 7

class *subset-extended* = *sup* + *inf* + *minus* + *uminus* + *bot* + *top* + *ord* +
assumes *sub-top-def*: $top = (THE\ x\ .\ \forall y\ .\ x = -y \sqcup --y)$
assumes *sub-bot-def*: $bot = -(THE\ x\ .\ \forall y\ .\ x = -y \sqcup --y)$
assumes *sub-inf-def*: $-x \sqcap -y = -(--x \sqcup --y)$

assumes *sub-minus-def*: $-x - -y = -(--x \sqcup -y)$
assumes *sub-less-eq-def*: $-x \leq -y \longleftrightarrow -x \sqcup -y = -y$
assumes *sub-less-def*: $-x < -y \longleftrightarrow -x \sqcup -y = -y \wedge \neg (-y \sqcup -x = -x)$

class *subset-boolean-algebra-extended* = *subset-boolean-algebra* + *subset-extended*
begin

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

consequences for meet

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

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

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

lemma *inf-idempotent*[*simp*]:
$-x \sqcap -x = -x$
⟨*proof*⟩

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

lemma *sup-absorb*[*simp*]:
$-x \sqcup (-x \sqcap -y) = -x$
⟨*proof*⟩

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

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

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

lemma *inf-cases*:

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

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

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

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

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

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

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

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

lemma *case-duality-2*:
$(-x \sqcap -y) \sqcup (--x \sqcap -z) = (-x \sqcup -z) \sqcap (--x \sqcup -y)$
⟨*proof*⟩

lemma *complement-cases*:
$((-v \sqcap -w) \sqcup (--v \sqcap -x)) \sqcap -((-v \sqcap -y) \sqcup (--v \sqcap -z)) = (-v \sqcap -w \sqcap --y) \sqcup (--v \sqcap -x \sqcap --z)$
⟨*proof*⟩

lemma *inf-cases-2*: $--x = -(-x \sqcap -y) \sqcap -(-x \sqcap --y)$
⟨*proof*⟩

consequences for *top* and *bot*

lemma *sup-complement*[*simp*]:
$-x \sqcup --x = top$
⟨*proof*⟩

lemma *inf-complement*[*simp*]:

$-x \sqcap --x = bot$
⟨*proof*⟩

lemma *complement-bot*[*simp*]:
$-bot = top$
⟨*proof*⟩

lemma *complement-top*[*simp*]:
$-top = bot$
⟨*proof*⟩

lemma *sup-right-zero*[*simp*]:
$-x \sqcup top = top$
⟨*proof*⟩

lemma *sup-left-zero*[*simp*]:
$top \sqcup -x = top$
⟨*proof*⟩

lemma *inf-right-unit*[*simp*]:
$-x \sqcap bot = bot$
⟨*proof*⟩

lemma *inf-left-unit*[*simp*]:
$bot \sqcap -x = bot$
⟨*proof*⟩

lemma *sup-right-unit*[*simp*]:
$-x \sqcup bot = -x$
⟨*proof*⟩

lemma *sup-left-unit*[*simp*]:
$bot \sqcup -x = -x$
⟨*proof*⟩

lemma *inf-right-zero*[*simp*]:
$-x \sqcap top = -x$
⟨*proof*⟩

lemma *sub-inf-left-zero*[*simp*]:
$top \sqcap -x = -x$
⟨*proof*⟩

lemma *bot-double-complement*[*simp*]:
$--bot = bot$
⟨*proof*⟩

lemma *top-double-complement*[*simp*]:
$--top = top$

⟨*proof*⟩

consequences for the order

lemma *reflexive*:
$-x \le -x$
⟨*proof*⟩

lemma *transitive*:
$-x \le -y \Longrightarrow -y \le -z \Longrightarrow -x \le -z$
⟨*proof*⟩

lemma *antisymmetric*:
$-x \le -y \Longrightarrow -y \le -x \Longrightarrow -x = -y$
⟨*proof*⟩

lemma *sub-bot-least*:
$bot \le -x$
⟨*proof*⟩

lemma *top-greatest*:
$-x \le top$
⟨*proof*⟩

lemma *upper-bound-left*:
$-x \le -x \sqcup -y$
⟨*proof*⟩

lemma *upper-bound-right*:
$-y \le -x \sqcup -y$
⟨*proof*⟩

lemma *sub-sup-left-isotone*:
assumes $-x \le -y$
shows $-x \sqcup -z \le -y \sqcup -z$
⟨*proof*⟩

lemma *sub-sup-right-isotone*:
$-x \le -y \Longrightarrow -z \sqcup -x \le -z \sqcup -y$
⟨*proof*⟩

lemma *sup-isotone*:
assumes $-p \le -q$
and $-r \le -s$
shows $-p \sqcup -r \le -q \sqcup -s$
⟨*proof*⟩

lemma *sub-complement-antitone*:
$-x \le -y \Longrightarrow --y \le --x$
⟨*proof*⟩

lemma *less-eq-inf*:
$-x \le -y \longleftrightarrow -x \sqcap -y = -x$
⟨*proof*⟩

lemma *inf-complement-left-antitone*:
$-x \le -y \Longrightarrow -(-y \sqcap -z) \le -(-x \sqcap -z)$
⟨*proof*⟩

lemma *sub-inf-left-isotone*:
$-x \le -y \Longrightarrow -x \sqcap -z \le -y \sqcap -z$
⟨*proof*⟩

lemma *sub-inf-right-isotone*:
$-x \le -y \Longrightarrow -z \sqcap -x \le -z \sqcap -y$
⟨*proof*⟩

lemma *inf-isotone*:
assumes $-p \le -q$
and $-r \le -s$
shows $-p \sqcap -r \le -q \sqcap -s$
⟨*proof*⟩

lemma *least-upper-bound*:
$-x \le -z \wedge -y \le -z \longleftrightarrow -x \sqcup -y \le -z$
⟨*proof*⟩

lemma *lower-bound-left*:
$-x \sqcap -y \le -x$
⟨*proof*⟩

lemma *lower-bound-right*:
$-x \sqcap -y \le -y$
⟨*proof*⟩

lemma *greatest-lower-bound*:
$-x \le -y \wedge -x \le -z \longleftrightarrow -x \le -y \sqcap -z$
⟨*proof*⟩

lemma *less-eq-sup-top*:
$-x \le -y \longleftrightarrow --x \sqcup -y = top$
⟨*proof*⟩

lemma *less-eq-inf-bot*:
$-x \le -y \longleftrightarrow -x \sqcap --y = bot$
⟨*proof*⟩

lemma *shunting*:
$-x \sqcap -y \le -z \longleftrightarrow -y \le --x \sqcup -z$

⟨*proof*⟩

lemma *shunting-right*:
$-x \sqcap -y \leq -z \longleftrightarrow -x \leq -z \sqcup --y$
⟨*proof*⟩

lemma *sup-less-eq-cases*:
assumes $-z \leq -x \sqcup -y$
and $-z \leq --x \sqcup -y$
shows $-z \leq -y$
⟨*proof*⟩

lemma *sup-less-eq-cases-2*:
$-x \sqcup -y \leq -x \sqcup -z \Longrightarrow --x \sqcup -y \leq --x \sqcup -z \Longrightarrow -y \leq -z$
⟨*proof*⟩

lemma *sup-less-eq-cases-3*:
$-y \sqcup -x \leq -z \sqcup -x \Longrightarrow -y \sqcup --x \leq -z \sqcup --x \Longrightarrow -y \leq -z$
⟨*proof*⟩

lemma *inf-less-eq-cases*:
$-x \sqcap -y \leq -z \Longrightarrow --x \sqcap -y \leq -z \Longrightarrow -y \leq -z$
⟨*proof*⟩

lemma *inf-less-eq-cases-2*:
$-x \sqcap -y \leq -x \sqcap -z \Longrightarrow --x \sqcap -y \leq --x \sqcap -z \Longrightarrow -y \leq -z$
⟨*proof*⟩

lemma *inf-less-eq-cases-3*:
$-y \sqcap -x \leq -z \sqcap -x \Longrightarrow -y \sqcap --x \leq -z \sqcap --x \Longrightarrow -y \leq -z$
⟨*proof*⟩

lemma *inf-eq-cases*:
$-x \sqcap -y = -x \sqcap -z \Longrightarrow --x \sqcap -y = --x \sqcap -z \Longrightarrow -y = -z$
⟨*proof*⟩

lemma *inf-eq-cases-2*:
$-y \sqcap -x = -z \sqcap -x \Longrightarrow -y \sqcap --x = -z \sqcap --x \Longrightarrow -y = -z$
⟨*proof*⟩

lemma *wnf-lemma-1*:
$((-x \sqcup -y) \sqcap (--x \sqcup -z)) \sqcup -x = -x \sqcup -y$
⟨*proof*⟩

lemma *wnf-lemma-2*:
$((-x \sqcup -y) \sqcap (-z \sqcup --y)) \sqcup -y = -x \sqcup -y$
⟨*proof*⟩

lemma *wnf-lemma-3*:

$((-x \sqcup -z) \sqcap (--x \sqcup -y)) \sqcup --x = --x \sqcup -y$
⟨*proof*⟩

lemma *wnf-lemma-4*:
$((-z \sqcup -y) \sqcap (-x \sqcup --y)) \sqcup --y = -x \sqcup --y$
⟨*proof*⟩

end

class *subset-boolean-algebra'* = *sup* + *uminus* +
assumes *sub-associative'*: $-x \sqcup (-y \sqcup -z) = (-x \sqcup -y) \sqcup -z$
assumes *sub-commutative'*: $-x \sqcup -y = -y \sqcup -x$
assumes *sub-complement'*: $-x = -(--x \sqcup -y) \sqcup -(--x \sqcup --y)$
assumes *sub-sup-closed'*: $\exists z . -x \sqcup -y = -z$
begin

subclass *subset-boolean-algebra*
⟨*proof*⟩

end

We introduce a type for the range of complement and show that it is an instance of *boolean-algebra*.

typedef (**overloaded**) *'a boolean-subset* = $\{ x::'a::uminus . \exists y . x = -y \}$
⟨*proof*⟩

lemma *simp-boolean-subset*[*simp*]:
$\exists y$. *Rep-boolean-subset* $x = -y$
⟨*proof*⟩

setup-lifting *type-definition-boolean-subset*

Theorem 8.1

instantiation *boolean-subset* :: (*subset-boolean-algebra*) *huntington*
begin

lift-definition *sup-boolean-subset* :: *'a boolean-subset* ⇒ *'a boolean-subset* ⇒ *'a boolean-subset* **is** *sup*
⟨*proof*⟩

lift-definition *uminus-boolean-subset* :: *'a boolean-subset* ⇒ *'a boolean-subset* **is** *uminus*
⟨*proof*⟩

instance
⟨*proof*⟩

end

Theorem 8.2

instantiation *boolean-subset* :: (*subset-boolean-algebra-extended*) *huntington-extended*
begin

lift-definition *inf-boolean-subset* :: $'a$ *boolean-subset* $\Rightarrow$ $'a$ *boolean-subset* $\Rightarrow$ $'a$ *boolean-subset* **is** *inf*
⟨*proof*⟩

lift-definition *minus-boolean-subset* :: $'a$ *boolean-subset* $\Rightarrow$ $'a$ *boolean-subset* $\Rightarrow$ $'a$ *boolean-subset* **is** *minus*
⟨*proof*⟩

lift-definition *bot-boolean-subset* :: $'a$ *boolean-subset* **is** *bot*
⟨*proof*⟩

lift-definition *top-boolean-subset* :: $'a$ *boolean-subset* **is** *top*
⟨*proof*⟩

lift-definition *less-eq-boolean-subset* :: $'a$ *boolean-subset* $\Rightarrow$ $'a$ *boolean-subset* $\Rightarrow$ *bool* **is** *less-eq* ⟨*proof*⟩

lift-definition *less-boolean-subset* :: $'a$ *boolean-subset* $\Rightarrow$ $'a$ *boolean-subset* $\Rightarrow$ *bool* **is** *less* ⟨*proof*⟩

instance
⟨*proof*⟩

end

5 Subset Boolean algebras with Additional Structure

We now discuss axioms that make the range of a unary operation a Boolean algebra, but add further properties that are common to the intended models. In the intended models, the unary operation can be a complement, a pseudocomplement or the antidomain operation. For simplicity, we mostly call the unary operation 'complement'.

We first look at structures based only on join and complement, and then add axioms for the remaining operations of Boolean algebras. In the intended models, the operation that is meet on the range of the complement can be a meet in the whole algebra or composition.

5.1 Axioms Derived from the New Axiomatisation

The axioms of the first algebra are based on *boolean-algebra-3*.

Definition 9

class *subset-boolean-algebra-1* = *sup* + *uminus* +
assumes *sba1-associative*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
assumes *sba1-commutative*: $x \sqcup y = y \sqcup x$
assumes *sba1-idempotent*[*simp*]: $x \sqcup x = x$
assumes *sba1-double-complement*[*simp*]: $---x = -x$
assumes *sba1-bot-unique*: $-(x \sqcup -x) = -(y \sqcup -y)$
assumes *sba1-export*: $-x \sqcup -(-x \sqcup y) = -x \sqcup -y$
begin

Theorem 11.1

subclass *subset-boolean-algebra*
⟨*proof*⟩

definition *sba1-bot* $\equiv$ *THE* x . $\forall z$. $x = -(z \sqcup -z)$

lemma *sba1-bot*:
sba1-bot $= -(z \sqcup -z)$
⟨*proof*⟩

end

Boolean algebra operations based on join and complement

Definition 10

class *subset-extended-1* = *sup* + *inf* + *minus* + *uminus* + *bot* + *top* + *ord* +
assumes *ba-bot*: *bot* = (*THE* x . $\forall z$. $x = -(z \sqcup -z)$)
assumes *ba-top*: *top* = $-$(*THE* x . $\forall z$. $x = -(z \sqcup -z)$)
assumes *ba-inf*: $-x \sqcap -y = -(--x \sqcup --y)$
assumes *ba-minus*: $-x - -y = -(--x \sqcup -y)$
assumes *ba-less-eq*: $x \leq y \longleftrightarrow x \sqcup y = y$
assumes *ba-less*: $x < y \longleftrightarrow x \sqcup y = y \wedge \neg (y \sqcup x = x)$

class *subset-extended-2* = *subset-extended-1* +
assumes *ba-bot-unique*: $-(x \sqcup -x) = -(y \sqcup -y)$
begin

lemma *ba-bot-def*:
bot $= -(z \sqcup -z)$
⟨*proof*⟩

lemma *ba-top-def*:
top $= --(z \sqcup -z)$
⟨*proof*⟩

end

Subset forms Boolean Algebra, extended by Boolean algebra operations

class *subset-boolean-algebra-1-extended* = *subset-boolean-algebra-1* +
subset-extended-1
begin

subclass *subset-extended-2*
⟨*proof*⟩

subclass *semilattice-sup*
⟨*proof*⟩

Theorem 11.2

subclass *subset-boolean-algebra-extended*
⟨*proof*⟩

end

5.2 Stronger Assumptions based on Join and Complement

We add further axioms covering properties common to the antidomain and (pseudo)complement instances.

Definition 12

class *subset-boolean-algebra-2* = *sup* + *uminus* +
assumes *sba2-associative*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
assumes *sba2-commutative*: $x \sqcup y = y \sqcup x$
assumes *sba2-idempotent*[*simp*]: $x \sqcup x = x$
assumes *sba2-bot-unit*: $x \sqcup -(y \sqcup -y) = x$
assumes *sba2-sub-sup-demorgan*: $-(x \sqcup y) = -(--x \sqcup --y)$
assumes *sba2-export*: $-x \sqcup -(-x \sqcup y) = -x \sqcup -y$
begin

Theorem 13.1

subclass *subset-boolean-algebra-1*
⟨*proof*⟩

Theorem 13.2

lemma *double-complement-dist-sup*:
$--(x \sqcup y) = --x \sqcup --y$
⟨*proof*⟩

lemma *maddux-3-3*[*simp*]:
$-(x \sqcup y) \sqcup -(x \sqcup -y) = -x$
⟨*proof*⟩

lemma *huntington-3-pp*[*simp*]:
$-(-x \sqcup -y) \sqcup -(-x \sqcup y) = --x$
⟨*proof*⟩

end

class *subset-boolean-algebra-2-extended* = *subset-boolean-algebra-2* + *subset-extended-1*

begin

subclass *subset-boolean-algebra-1-extended* ⟨*proof*⟩

subclass *bounded-semilattice-sup-bot*
⟨*proof*⟩

Theorem 13.3

lemma *complement-antitone*:
$x \leq y \Longrightarrow -y \leq -x$
⟨*proof*⟩

lemma *double-complement-isotone*:
$x \leq y \Longrightarrow --x \leq --y$
⟨*proof*⟩

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

end

5.3 Axioms for Meet

We add further axioms of *inf* covering properties common to the antidomain and pseudocomplement instances. We omit the left distributivity rule and the right zero rule as they do not hold in some models. In particular, the operation *inf* does not have to be commutative.

Definition 14

class *subset-boolean-algebra-3-extended* = *subset-boolean-algebra-2-extended* +
assumes *sba3-inf-associative*: $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$
assumes *sba3-inf-right-dist-sup*: $(x \sqcup y) \sqcap z = (x \sqcap z) \sqcup (y \sqcap z)$
assumes *sba3-inf-complement-bot*: $-x \sqcap x = bot$
assumes *sba3-inf-left-unit*[*simp*]: $top \sqcap x = x$
assumes *sba3-complement-inf-double-complement*: $-(x \sqcap --y) = -(x \sqcap y)$
begin

Theorem 15

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

lemma *inf-double-complement-export*:
$--(--x \sqcap y) = --x \sqcap --y$
⟨*proof*⟩

lemma *inf-left-isotone*:
$x \leq y \Longrightarrow x \sqcap z \leq y \sqcap z$

⟨*proof*⟩

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

lemma *double-complement-above*:
$--x \sqcap x = x$
⟨*proof*⟩

lemma $x \le y \Longrightarrow z \sqcap x \le z \sqcap y$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma $x \sqcap top = x$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma $x \sqcap y = y \sqcap x$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

5.4 Stronger Assumptions for Meet

The following axioms also hold in both models, but follow from the axioms of *subset-boolean-algebra-5-operations*.

Definition 16

class *subset-boolean-algebra-4-extended* = *subset-boolean-algebra-3-extended* +
assumes *sba4-inf-right-unit*[*simp*]: $x \sqcap top = x$
assumes *inf-right-isotone*: $x \le y \Longrightarrow z \sqcap x \le z \sqcap y$
begin

lemma $x \sqcup top = top$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma $x \sqcap bot = bot$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma $(x \sqcap y = bot) = (x \le -\ y)$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

6 Boolean Algebras in Stone Algebras

We specialise *inf* to meet and complement to pseudocomplement. This puts Stone algebras into the picture; for these it is well known that regular elements form a Boolean subalgebra [12].

Definition 17

class *subset-boolean-algebra-5-extended* = *subset-boolean-algebra-3-extended* +
assumes *sba5-inf-commutative*: $x \sqcap y = y \sqcap x$
assumes *sba5-inf-absorb*: $x \sqcap (x \sqcup y) = x$
begin

subclass *distrib-lattice-bot*
⟨*proof*⟩

lemma *inf-demorgan-2*:
$-(x \sqcap y) = -x \sqcup -y$
⟨*proof*⟩

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

lemma *complement-inf*[*simp*]:
$x \sqcap -x = bot$
⟨*proof*⟩

Theorem 18.2

subclass *stone-algebra*
⟨*proof*⟩

Theorem 18.1

subclass *subset-boolean-algebra-4-extended*
⟨*proof*⟩

end

context *stone-algebra-extended*
begin

Theorem 18.3

subclass *subset-boolean-algebra-5-extended*
⟨*proof*⟩

end

7 Domain Semirings

The following development of tests in IL-semirings, prepredomain semirings, predomain semirings and domain semirings is mostly based on [23]; see also [4]. See [5] for domain axioms in idempotent semirings. See [3, 19] for domain axioms in semigroups and monoids. Some variants have been implemented in [11].

7.1 Idempotent Left Semirings

Definition 19

class *il-semiring* = *sup* + *inf* + *bot* + *top* + *ord* +
assumes *il-associative*: $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
assumes *il-commutative*: $x \sqcup y = y \sqcup x$
assumes *il-idempotent*[*simp*]: $x \sqcup x = x$

assumes *il-bot-unit*: $x \sqcup bot = x$
assumes *il-inf-associative*: $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$
assumes *il-inf-right-dist-sup*: $(x \sqcup y) \sqcap z = (x \sqcap z) \sqcup (y \sqcap z)$
assumes *il-inf-left-unit*[*simp*]: $top \sqcap x = x$
assumes *il-inf-right-unit*[*simp*]: $x \sqcap top = x$
assumes *il-sub-inf-left-zero*[*simp*]: $bot \sqcap x = bot$
assumes *il-sub-inf-right-isotone*: $x \leq y \Longrightarrow z \sqcap x \leq z \sqcap y$
assumes *il-less-eq*: $x \leq y \longleftrightarrow x \sqcup y = y$
assumes *il-less-def*: $x < y \longleftrightarrow x \leq y \wedge \neg(y \leq x)$
begin

lemma *il-unit-bot*: $bot \sqcup x = x$
⟨*proof*⟩

subclass *order*
⟨*proof*⟩

lemma *il-sub-inf-right-isotone-var*:
$(x \sqcap y) \sqcup (x \sqcap z) \leq x \sqcap (y \sqcup z)$
⟨*proof*⟩

lemma *il-sub-inf-left-isotone*:
$x \leq y \Longrightarrow x \sqcap z \leq y \sqcap z$
⟨*proof*⟩

lemma *il-sub-inf-left-isotone-var*:
$(y \sqcap x) \sqcup (z \sqcap x) \leq (y \sqcup z) \sqcap x$
⟨*proof*⟩

lemma *sup-left-isotone*:
$x \leq y \Longrightarrow x \sqcup z \leq y \sqcup z$
⟨*proof*⟩

lemma *sup-right-isotone*:
$x \leq y \Longrightarrow z \sqcup x \leq z \sqcup y$
⟨*proof*⟩

lemma *bot-least*:
$bot \leq x$
⟨*proof*⟩

lemma *less-eq-bot*:
$x \leq bot \longleftrightarrow x = bot$
⟨*proof*⟩

abbreviation *are-complementary* :: $'a \Rightarrow 'a \Rightarrow bool$
where *are-complementary* $x\ y \equiv x \sqcup y = top \wedge x \sqcap y = bot \wedge y \sqcap x = bot$

abbreviation *test* :: $'a \Rightarrow bool$

where *test* $x \equiv \exists\, y$. *are-complementary* $x\ y$

definition *tests* :: *'a set*
where *tests* = { x . *test* x }

lemma *bot-test*:
test bot
⟨*proof*⟩

lemma *top-test*:
test top
⟨*proof*⟩

lemma *test-sub-identity*:
test $x \Longrightarrow x \leq$ *top*
⟨*proof*⟩

lemma *neg-unique*:
are-complementary $x\ y \Longrightarrow$ *are-complementary* $x\ z \Longrightarrow y = z$
⟨*proof*⟩

definition *neg* :: *'a* ⇒ *'a* (‹!›)
where $!x \equiv$ *THE* y . *are-complementary* $x\ y$

lemma *neg-char*:
assumes *test* x
shows *are-complementary* x $(!x)$
⟨*proof*⟩

lemma *are-complementary-symmetric*:
are-complementary $x\ y \longleftrightarrow$ *are-complementary* $y\ x$
⟨*proof*⟩

lemma *neg-test*:
test $x \Longrightarrow$ *test* $(!x)$
⟨*proof*⟩

lemma *are-complementary-test*:
test $x \Longrightarrow$ *are-complementary* $x\ y \Longrightarrow$ *test* y
⟨*proof*⟩

lemma *neg-involutive*:
test $x \Longrightarrow\ !(!x) = x$
⟨*proof*⟩

lemma *test-inf-left-below*:
test $x \Longrightarrow x \sqcap y \leq y$
⟨*proof*⟩

lemma *test-inf-right-below*:
  $test\ x \Longrightarrow y \sqcap x \leq y$
  ⟨*proof*⟩

lemma *neg-bot*:
  $!bot = top$
  ⟨*proof*⟩

lemma *neg-top*:
  $!top = bot$
  ⟨*proof*⟩

lemma *test-inf-idempotent*:
  $test\ x \Longrightarrow x \sqcap x = x$
  ⟨*proof*⟩

lemma *test-inf-semicommutative*:
  assumes $test\ x$
    and $test\ y$
  shows $x \sqcap y \leq y \sqcap x$
⟨*proof*⟩

lemma *test-inf-commutative*:
  $test\ x \Longrightarrow test\ y \Longrightarrow x \sqcap y = y \sqcap x$
  ⟨*proof*⟩

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

lemma *test-absorb-1*:
  $test\ x \Longrightarrow test\ y \Longrightarrow x \sqcup (x \sqcap y) = x$
  ⟨*proof*⟩

lemma *test-absorb-2*:
  $test\ x \Longrightarrow test\ y \Longrightarrow x \sqcup (y \sqcap x) = x$
  ⟨*proof*⟩

lemma *test-absorb-3*:
  $test\ x \Longrightarrow test\ y \Longrightarrow x \sqcap (x \sqcup y) = x$
  ⟨*proof*⟩

lemma *test-absorb-4*:
  $test\ x \Longrightarrow test\ y \Longrightarrow (x \sqcup y) \sqcap x = x$
  ⟨*proof*⟩

lemma *test-import-1*:
  assumes $test\ x$
    and $test\ y$

shows $x \sqcup (!x \sqcap y) = x \sqcup y$
⟨*proof*⟩

lemma *test-import-2*:
assumes *test* x
and *test* y
shows $x \sqcup (y \sqcap !x) = x \sqcup y$
⟨*proof*⟩

lemma *test-import-3*:
assumes *test* x
shows $(!x \sqcup y) \sqcap x = y \sqcap x$
⟨*proof*⟩

lemma *test-import-4*:
assumes *test* x
and *test* y
shows $(!x \sqcup y) \sqcap x = x \sqcap y$
⟨*proof*⟩

lemma *test-inf*:
test $x \Longrightarrow$ *test* $y \Longrightarrow$ *test* $z \Longrightarrow z \leq x \sqcap y \longleftrightarrow z \leq x \wedge z \leq y$
⟨*proof*⟩

lemma *test-shunting*:
assumes *test* x
and *test* y
shows $x \sqcap y \leq z \longleftrightarrow x \leq !y \sqcup z$
⟨*proof*⟩

lemma *test-shunting-bot*:
assumes *test* x
and *test* y
shows $x \leq y \longleftrightarrow x \sqcap !y \leq bot$
⟨*proof*⟩

lemma *test-shunting-bot-eq*:
assumes *test* x
and *test* y
shows $x \leq y \longleftrightarrow x \sqcap !y = bot$
⟨*proof*⟩

lemma *neg-antitone*:
assumes *test* x
and *test* y
and $x \leq y$
shows $!y \leq !x$
⟨*proof*⟩

lemma *test-sup-neg-1*:
assumes *test* x
and *test* y
shows $(x \sqcup y) \sqcup (!x \sqcap !y) = top$
⟨*proof*⟩

lemma *test-sup-neg-2*:
assumes *test* x
and *test* y
shows $(x \sqcup y) \sqcap (!x \sqcap !y) = bot$
⟨*proof*⟩

lemma *de-morgan-1*:
assumes *test* x
and *test* y
and *test* $(x \sqcap y)$
shows $!(x \sqcap y) = !x \sqcup !y$
⟨*proof*⟩

lemma *de-morgan-2*:
assumes *test* x
and *test* y
and *test* $(x \sqcup y)$
shows $!(x \sqcup y) = !x \sqcap !y$
⟨*proof*⟩

lemma *test-inf-closed-sup-complement*:
assumes *test* x
and *test* y
and $\forall\, u\ v\ .\ test\ u \wedge test\ v \longrightarrow test\ (u \sqcap v)$
shows $!x \sqcap !y \sqcap (x \sqcup y) = bot$
⟨*proof*⟩

lemma *test-sup-complement-sup-closed*:
assumes *test* x
and *test* y
and $\forall\, u\ v\ .\ test\ u \wedge test\ v \longrightarrow !u \sqcap !v \sqcap (u \sqcup v) = bot$
shows *test* $(x \sqcup y)$
⟨*proof*⟩

lemma *test-inf-closed-sup-closed*:
assumes *test* x
and *test* y
and $\forall\, u\ v\ .\ test\ u \wedge test\ v \longrightarrow test\ (u \sqcap v)$
shows *test* $(x \sqcup y)$
⟨*proof*⟩

end

7.2 Prepredomain Semirings

class *dom* =
  fixes $d :: 'a \Rightarrow 'a$

class *ppd-semiring* = *il-semiring* + *dom* +
  assumes *d-closed*: $test\ (d\ x)$
  assumes *d1*: $x \leq d\ x \sqcap x$
begin

lemma *d-sub-identity*:
  $d\ x \leq top$
  ⟨*proof*⟩

lemma *d1-eq*:
  $x = d\ x \sqcap x$
⟨*proof*⟩

lemma *d-increasing-sub-identity*:
  $x \leq top \Longrightarrow x \leq d\ x$
  ⟨*proof*⟩

lemma *d-top*:
  $d\ top = top$
  ⟨*proof*⟩

lemma *d-bot-only*:
  $d\ x = bot \Longrightarrow x = bot$
  ⟨*proof*⟩

lemma *d-strict*: $d\ bot \leq bot$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma *d-isotone-var*: $d\ x \leq d\ (x \sqcup y)$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma *d-fully-strict*: $d\ x = bot \longleftrightarrow x = bot$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma *test-d-fixpoint*: $test\ x \Longrightarrow d\ x = x$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

7.3 Predomain Semirings

class *pd-semiring* = *ppd-semiring* +
  assumes *d2*: $test\ p \Longrightarrow d\ (p \sqcap x) \leq p$
begin

lemma *d-strict*:
  $d\ bot \leq bot$
  ⟨*proof*⟩

lemma *d-strict-eq*:
  $d\ bot = bot$
  ⟨*proof*⟩

lemma *test-d-fixpoint*:
$test\ x \Longrightarrow d\ x = x$
⟨*proof*⟩

lemma *d-surjective*:
$test\ x \Longrightarrow \exists\, y\ .\ d\ y = x$
⟨*proof*⟩

lemma *test-d-fixpoint-iff*:
$test\ x \longleftrightarrow d\ x = x$
⟨*proof*⟩

lemma *d-surjective-iff*:
$test\ x \longleftrightarrow (\exists\, y\ .\ d\ y = x)$
⟨*proof*⟩

lemma *tests-d-range*:
$tests = range\ d$
⟨*proof*⟩

lemma *llp*:
assumes $test\ y$
shows $d\ x \le y \longleftrightarrow x \le y \sqcap x$
⟨*proof*⟩

lemma *gla*:
assumes $test\ y$
shows $y \le !(d\ x) \longleftrightarrow y \sqcap x \le bot$
⟨*proof*⟩

lemma *gla-var*:
$test\ y \Longrightarrow y \sqcap d\ x \le bot \longleftrightarrow y \sqcap x \le bot$
⟨*proof*⟩

lemma *llp-var*:
assumes $test\ y$
shows $y \le !(d\ x) \longleftrightarrow x \le !y \sqcap x$
⟨*proof*⟩

lemma *d-idempotent*:
$d\ (d\ x) = d\ x$
⟨*proof*⟩

lemma *d-neg*:
$test\ x \Longrightarrow d\ (!x) = !x$
⟨*proof*⟩

lemma *d-fully-strict*:

$d\ x = bot \longleftrightarrow x = bot$
⟨*proof*⟩

lemma *d-ad-comp*:
$!(d\ x) \sqcap x = bot$
⟨*proof*⟩

lemma *d-isotone*:
assumes $x \leq y$
shows $d\ x \leq d\ y$
⟨*proof*⟩

lemma *d-isotone-var*:
$d\ x \leq d\ (x \sqcup y)$
⟨*proof*⟩

lemma *d3-conv*:
$d\ (x \sqcap y) \leq d\ (x \sqcap d\ y)$
⟨*proof*⟩

lemma *d-test-inf-idempotent*:
$d\ x \sqcap d\ x = d\ x$
⟨*proof*⟩

lemma *d-test-inf-closed*:
assumes *test x*
and *test y*
shows $d\ (x \sqcap y) = x \sqcap y$
⟨*proof*⟩

lemma *test-inf-closed*:
$test\ x \Longrightarrow test\ y \Longrightarrow test\ (x \sqcap y)$
⟨*proof*⟩

lemma *test-sup-closed*:
$test\ x \Longrightarrow test\ y \Longrightarrow test\ (x \sqcup y)$
⟨*proof*⟩

lemma *d-export*:
assumes *test x*
shows $d\ (x \sqcap y) = x \sqcap d\ y$
⟨*proof*⟩

lemma *test-inf-left-dist-sup*:
assumes *test x*
and *test y*
and *test z*
shows $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$
⟨*proof*⟩

lemma $!x \sqcup !y = !(!(!x \sqcup !y))$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma $d\ x = !(!x)$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

sublocale *subset-boolean-algebra* **where** $uminus = \lambda\ x\ .\ !(d\ x)$
⟨*proof*⟩

lemma *d-dist-sup*:
$d\ (x \sqcup y) = d\ x \sqcup d\ y$
⟨*proof*⟩

end

class *pd-semiring-extended* = *pd-semiring* + *uminus* +
assumes *uminus-def*: $-x = !(d\ x)$
begin

subclass *subset-boolean-algebra*
⟨*proof*⟩

end

7.4 Domain Semirings

class *d-semiring* = *pd-semiring* +
assumes *d3*: $d\ (x \sqcap d\ y) \le d\ (x \sqcap y)$
begin

lemma *d3-eq*: $d\ (x \sqcap d\ y) = d\ (x \sqcap y)$
⟨*proof*⟩

end

Axioms (d1), (d2) and (d3) are independent in IL-semirings.

context *il-semiring*
begin

context
fixes $d :: 'a \Rightarrow 'a$
assumes *d-closed*: *test* $(d\ x)$
begin

context
assumes *d1*: $x \le d\ x \sqcap x$
assumes *d2*: *test* $p \Longrightarrow d\ (p \sqcap x) \le p$
begin

lemma *d3*: $d\ (x \sqcap d\ y) \le d\ (x \sqcap y)$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

context
  assumes *d1*: $x \leq d\ x \sqcap x$
  assumes *d3*: $d\ (x \sqcap d\ y) \leq d\ (x \sqcap y)$
begin

lemma *d2*: $test\ p \Longrightarrow d\ (p \sqcap x) \leq p$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

context
  assumes *d2*: $test\ p \Longrightarrow d\ (p \sqcap x) \leq p$
  assumes *d3*: $d\ (x \sqcap d\ y) \leq d\ (x \sqcap y)$
begin

lemma *d1*: $x \leq d\ x \sqcap x$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

end

end

class *d-semiring-var* = *ppd-semiring* +
  assumes *d3-var*: $d\ (x \sqcap d\ y) \leq d\ (x \sqcap y)$
  assumes *d-strict-eq-var*: $d\ bot = bot$
begin

lemma *d2-var*:
  assumes *test p*
    shows $d\ (p \sqcap x) \leq p$
⟨*proof*⟩

subclass *d-semiring*
⟨*proof*⟩

end

8 Antidomain Semirings

We now develop prepreantidomain semirings, preantidomain semirings and antidomain semirings. See [6, 7, 8] for related work on internal axioms for antidomain.

8.1 Prepreantidomain Semirings

Definition 20

class *ppa-semiring* = *il-semiring* + *uminus* +

assumes *a-inf-complement-bot*: $-x \sqcap x = bot$
assumes *a-stone*[*simp*]: $-x \sqcup --x = top$
begin

Theorem 21

lemma *l1*:
$-top = bot$
⟨*proof*⟩

lemma *l2*:
$-bot = top$
⟨*proof*⟩

lemma *l3*:
$-x \leq -y \Longrightarrow -x \sqcap y = bot$
⟨*proof*⟩

lemma *l5*:
$--x \leq --y \Longrightarrow -y \leq -x$
⟨*proof*⟩

lemma *l4*:
$---x = -x$
⟨*proof*⟩

lemma *l6*:
$-x \sqcap --x = bot$
⟨*proof*⟩

lemma *l7*:
$-x \sqcap -y = -y \sqcap -x$
⟨*proof*⟩

lemma *l8*:
$x \leq --x \sqcap x$
⟨*proof*⟩

sublocale *ppa-ppd*: *ppd-semiring* **where** $d = \lambda x . --x$
⟨*proof*⟩

end

8.2 Preantidomain Semirings

Definition 22

class *pa-semiring* = *ppa-semiring* +
assumes *pad2*: $--x \leq -(-x \sqcap y)$

begin

Theorem 23

lemma *l10*:
$-x \sqcap y = bot \Longrightarrow -x \leq -y$
⟨*proof*⟩

lemma *l10-iff*:
$-x \sqcap y = bot \longleftrightarrow -x \leq -y$
⟨*proof*⟩

lemma *l13*:
$--(--x \sqcap y) \leq --x$
⟨*proof*⟩

lemma *l14*:
$-(x \sqcap --y) \leq -(x \sqcap y)$
⟨*proof*⟩

lemma *l9*:
$x \leq y \Longrightarrow -y \leq -x$
⟨*proof*⟩

lemma *l11*:
$- x \sqcup - y = - (- - x \sqcap - - y)$
⟨*proof*⟩

lemma *l12*:
$- x \sqcap - y = - (x \sqcup y)$
⟨*proof*⟩

lemma *l15*:
$--(x \sqcup y) = --x \sqcup --y$
⟨*proof*⟩

lemma *l13-var*:
$- - (- x \sqcap y) = - x \sqcap - - y$
⟨*proof*⟩

Theorem 25.1

subclass *subset-boolean-algebra-2*
⟨*proof*⟩

lemma *aa-test*:
$p = --p \Longrightarrow test\ p$
⟨*proof*⟩

lemma *test-aa-increasing*:
$test\ p \Longrightarrow p \leq --p$

⟨*proof*⟩

lemma *test* $p \Longrightarrow - - (p \sqcap x) \leq p$ **nitpick** [*expect=genuine*] ⟨*proof*⟩
lemma *test* $p \Longrightarrow --p \leq p$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

class *pa-algebra* = *pa-semiring* + *minus* +
assumes *pa-minus-def*: $-x - -y = -(--x \sqcup -y)$
begin

subclass *subset-boolean-algebra-2-extended*
⟨*proof*⟩

lemma $\bigwedge x\ y.\ - (x \sqcap - - y) = - (x \sqcap y)$ **nitpick** [*expect=genuine*] ⟨*proof*⟩

end

8.3 Antidomain Semirings

Definition 24

class *a-semiring* = *ppa-semiring* +
assumes *ad3*: $-(x \sqcap y) \leq -(x \sqcap --y)$
begin

lemma *l16*:
$- - x \leq - (- x \sqcap y)$
⟨*proof*⟩

Theorem 25.2

subclass *pa-semiring*
⟨*proof*⟩

lemma *l17*:
$-(x \sqcap y) = -(x \sqcap --y)$
⟨*proof*⟩

lemma *a-complement-inf-double-complement*:
$-(x \sqcap --y) = -(x \sqcap y)$
⟨*proof*⟩

sublocale *a-d*: *d-semiring-var* **where** $d = \lambda x\ .\ --x$
⟨*proof*⟩

lemma *test* $p \Longrightarrow - - (p \sqcap x) \leq p$
⟨*proof*⟩

end

class *a-algebra* = *a-semiring* + *minus* +
  assumes *a-minus-def*: $-x - -y = -(--x \sqcup -y)$
begin

subclass *pa-algebra*
⟨*proof*⟩

Theorem 25.4

subclass *subset-boolean-algebra-4-extended*
⟨*proof*⟩

end

context *subset-boolean-algebra-4-extended*
begin

subclass *il-semiring*
⟨*proof*⟩

subclass *a-semiring*
⟨*proof*⟩

sublocale *sba4-a*: *a-algebra*
⟨*proof*⟩

end

context *stone-algebra*
begin

Theorem 25.3

subclass *il-semiring*
⟨*proof*⟩

subclass *a-semiring*
⟨*proof*⟩

end

end

References

[1] R. Balbes and A. Horn. Stone lattices. *Duke Mathematical Journal*, 37(3):537–545, 1970.

[2] L. Byrne. Two brief formulations of Boolean algebra. *Bulletin of the American Mathematical Society*, 52(4):269–272, 1946.

[3] J. Desharnais, P. Jipsen, and G. Struth. Domain and antidomain semigroups. In R. Berghammer, A. M. Jaoua, and B. Möller, editors, *Relations and Kleene Algebra in Computer Science (RelMiCS/AKA 2009)*, volume 5827 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2009.

[4] J. Desharnais and B. Möller. Fuzzifying modal algebra. In P. Höfner, P. Jipsen, W. Kahl, and M. E. Müller, editors, *Relational and Algebraic Methods in Computer Science (RAMiCS 2014)*, volume 8428 of *Lecture Notes in Computer Science*, pages 395–411. Springer, 2014.

[5] J. Desharnais, B. Möller, and G. Struth. Kleene algebra with domain. *ACM Transactions on Computational Logic*, 7(4):798–833, 2006.

[6] J. Desharnais and G. Struth. Domain axioms for a family of near-semirings. In J. Meseguer and G. Roşu, editors, *Algebraic Methodology and Software Technology (AMAST 2008)*, volume 5140 of *Lecture Notes in Computer Science*, pages 330–345. Springer, 2008.

[7] J. Desharnais and G. Struth. Modal semirings revisited. In P. Audebaud and C. Paulin-Mohring, editors, *Mathematics of Program Construction (MPC 2008)*, volume 5133 of *Lecture Notes in Computer Science*, pages 360–387. Springer, 2008.

[8] J. Desharnais and G. Struth. Internal axioms for domain semirings. *Sci. Comput. Programming*, 76(3):181–203, 2011.

[9] O. Frink. Pseudo-complements in semi-lattices. *Duke Mathematical Journal*, 29(4):505–514, 1962.

[10] O. Frink, Jr. Representations of Boolean algebras. *Bulletin of the American Mathematical Society*, 47(10):755–756, 1941.

[11] V. B. F. Gomes, W. Guttmann, P. Höfner, G. Struth, and T. Weber. Kleene algebras with domain. *Archive of Formal Proofs*, 2016.

[12] G. Grätzer. *Lattice Theory: First Concepts and Distributive Lattices.* W. H. Freeman and Co., 1971.

[13] W. Guttmann. Algebras for iteration and infinite computations. *Acta Inf.*, 49(5):343–359, 2012.

[14] W. Guttmann. Verifying minimum spanning tree algorithms with Stone relation algebras. *Journal of Logical and Algebraic Methods in Programming*, 101:132–150, 2018.

[15] W. Guttmann and B. Möller. A hierarchy of algebras for Boolean subsets. In U. Fahrenberg, P. Jipsen, and M. Winter, editors, *Relational*

and Algebraic Methods in Computer Science (RAMiCS 2020), volume 12062 of *Lecture Notes in Computer Science*, pages 152–168. Springer, 2020.

[16] W. Guttmann, G. Struth, and T. Weber. Automating algebraic methods in Isabelle. In S. Qin and Z. Qiu, editors, *Formal Methods and Software Engineering (ICFEM 2011)*, volume 6991 of *Lecture Notes in Computer Science*, pages 617–632. Springer, 2011.

[17] M. Hollenberg. An equational axiomatization of dynamic negation and relational composition. *Journal of Logic, Language, and Information*, 6(4):381–401, 1997.

[18] E. V. Huntington. Boolean algebra. A correction. *Transactions of the American Mathematical Society*, 35(2):557–558, 1933.

[19] M. Jackson and T. Stokes. Semilattice pseudo-complements on semigroups. *Communications in Algebra*, 32(8):2895–2918, 2004.

[20] R. D. Maddux. Relation-algebraic semantics. *Theoretical Comput. Sci.*, 160(1–2):1–85, 1996.

[21] W. McCune. Prover9 and Mace4. Accessed 14 January 2020 at `https://www.cs.unm.edu/~mccune/prover9/`, 2005–2010.

[22] C. A. Meredith and A. N. Prior. Equational logic. *Notre Dame Journal of Formal Logic*, 9(3):212–226, 1968.

[23] B. Möller and J. Desharnais. Basics of modal semirings and of Kleene/omega algebras. Report 2019-03, Institut für Informatik, Universität Augsburg, 2019.

[24] M. Wampler-Doty. A complete proof of the Robbins conjecture. *Archive of Formal Proofs*, 2016, first version 2010.