

# Topological semantics for paraconsistent and paracomplete logics

David Fuenmayor

March 3, 2021

## Abstract

We introduce a generalized topological semantics for paraconsistent and paracomplete logics by drawing upon early works on topological Boolean algebras (cf. works by Kuratowski, Zarycki, McKinsey & Tarski, etc.). In particular, this work exemplarily illustrates the shallow semantical embeddings approach (SSE) employing the proof assistant Isabelle/HOL. By means of the SSE technique we can effectively harness theorem provers, model finders and ‘hammers’ for reasoning with quantified non-classical logics.

## Contents

<b>1</b>	<b>Shallow embedding of a Boolean algebra of propositions</b>	<b>3</b>
1.1	Encoding Boolean operations . . . . .	4
1.2	Second-order operations and fixed-points . . . . .	4
1.3	Equality and atomicity . . . . .	5
1.4	Obtaining a complete Boolean Algebra . . . . .	6
1.5	Adding quantifiers (restricted and unrestricted) . . . . .	6
1.6	Relating quantifiers with further operators . . . . .	7
<b>2</b>	<b>Positive semantic conditions for operations</b>	<b>8</b>
2.1	Definitions (finitary case) . . . . .	9
2.2	Relations among conditions (finitary case) . . . . .	9
2.3	Definitions (infinitary case) . . . . .	10
2.4	Relations among conditions (infinitary case) . . . . .	10
2.5	Exploring the Barcan formula and its converse . . . . .	10
<b>3</b>	<b>Negative semantic conditions for operations</b>	<b>11</b>
3.1	Definitions and interrelations (finitary case) . . . . .	11
3.1.1	Principles of excluded middle, contradiction and explosion . . . . .	11
3.1.2	Contraposition rules . . . . .	12
3.1.3	Modus tollens rules . . . . .	12
3.1.4	Double negation introduction and elimination . . . . .	13
3.1.5	Normality and its dual . . . . .	14
3.1.6	De Morgan laws . . . . .	14
3.1.7	Contextual (strong) contraposition rule . . . . .	14
3.1.8	Local contraposition axioms . . . . .	15
3.1.9	Local modus tollens axioms . . . . .	16
3.1.10	Disjunctive syllogism . . . . .	16
3.2	Definitions and interrelations (infinitary case) . . . . .	17

<b>4</b>	<b>Topological operators</b>	<b>18</b>
4.1	Interior and closure . . . . .	18
4.1.1	Interior conditions . . . . .	19
4.1.2	Closure conditions . . . . .	19
4.1.3	Exploring dualities . . . . .	20
4.2	Frontier and border . . . . .	20
4.2.1	Frontier conditions . . . . .	20
4.2.2	Border conditions . . . . .	20
4.2.3	Relation with closure and interior . . . . .	21
4.2.4	Infinitary conditions . . . . .	23
4.3	Derivative and coherence . . . . .	23
4.3.1	Derivative conditions . . . . .	23
4.3.2	Infinitary conditions . . . . .	25
4.3.3	Coherence conditions . . . . .	26
<b>5</b>	<b>Generalized specialization orderings and Alexandrov topologies</b>	<b>26</b>
5.1	Specialization relations . . . . .	27
5.2	Alexandrov topology . . . . .	28
5.3	(Anti)symmetry and the separation axioms T0 and T1 . . . . .	28
<b>6</b>	<b>Frontier Algebra</b>	<b>29</b>
6.1	Basic properties . . . . .	29
6.2	Further properties . . . . .	31
<b>7</b>	<b>Frontier-based negation - Semantic conditions</b>	<b>32</b>
7.1	'Explosion' (ECQ), non-contradiction (LNC) and excluded middle (TND) . . . . .	32
7.2	Modus tollens (MT) . . . . .	33
7.3	Contraposition rules (CoP) . . . . .	33
7.4	Normality (negative) and its dual (nNor/nDNor) . . . . .	34
7.5	Double negation introduction/elimination (DNI/DNE) . . . . .	34
7.6	De Morgan laws . . . . .	35
7.7	Local contraposition axioms (lCoP) . . . . .	36
7.8	Disjunctive syllogism . . . . .	37
<b>8</b>	<b>Frontier-based negation - Fixed-points</b>	<b>37</b>
8.1	'Explosion' (ECQ) and excluded middle (TND) . . . . .	38
8.2	Contraposition rules . . . . .	38
8.3	Double negation introduction/elimination . . . . .	39
8.4	De Morgan laws . . . . .	39
8.5	Local contraposition axioms . . . . .	40
8.6	Disjunctive syllogism . . . . .	40
<b>9</b>	<b>Example application: Logics of Formal Inconsistency (LFIs)</b>	<b>41</b>
<b>10</b>	<b>Strict implication</b>	<b>44</b>
<b>11</b>	<b>Example application: Subintuitionistic and subminimal logics</b>	<b>46</b>
<b>12</b>	<b>Derivative algebra</b>	<b>50</b>
12.1	Basic properties . . . . .	50
12.2	Further properties . . . . .	51

<b>13 Example application: Logics of Formal Undeterminedness (LFUs)</b>	<b>52</b>
<b>14 Border algebra</b>	<b>54</b>
14.1 Basic properties . . . . .	54
<b>15 Closure algebra</b>	<b>55</b>
15.1 Basic properties . . . . .	55
<b>16 Interior algebra</b>	<b>56</b>
16.1 Basic properties . . . . .	56

```
theory sse-boolean-algebra
  imports Main
begin
```

```
declare[[syntax-ambiguity-warning=false]]
nitpick-params[assms=true, user-axioms=true, show-all, expect=genuine, format=3]
```

## 1 Shallow embedding of a Boolean algebra of propositions

In this section we present a shallow semantical embedding (SSE, cf. [1] and [2]) for a family of logics whose semantics is based upon extensions of (complete and atomic) Boolean algebras. The range of such logics is indeed very wide, including, as we will see, quantified paraconsistent and paracomplete (e.g. intuitionistic) logics. Aside from illustrating how the SSE approach works in practice we show how it allows us to effectively harness theorem provers, model finders and ‘hammers’ for reasoning with quantified non-classical logics. Proof reconstructions have deliberately been avoided. Most of the proofs (in fact, all one-liners) have been found using Sledgehammer.

Two notions play a fundamental role in this work: propositions and propositional functions. Propositions, qua sentence denotations, are modeled as objects of type  $w \Rightarrow bool$  (shortened as  $\sigma$ ). Propositional functions, as the name indicates, are basically anything with a (parametric) type  $t \Rightarrow \sigma$ .

We introduce a type  $w$  for the domain of points (aka. ‘worlds’, ‘states’, etc.).  $\sigma$  is a type alias for sets of points (i.e. propositions) encoded as characteristic functions.

```
typedecl w
type-synonym  $\sigma = w \Rightarrow bool$ 
```

In the sequel, we introduce the following naming convention for variables:

- (i) Latin letters (A, b, M, P, q, X, y, etc.) denote in general propositions (type  $\sigma$ ); however, we reserve letters D and S to denote sets of propositions (aka. domains/spaces) and the letters u, v and w to denote worlds/points.
- (ii) Greek letters (in particular  $\pi$ ) denote propositional functions (type  $t \Rightarrow \sigma$ ); among the latter we may employ the letters  $\varphi$ ,  $\psi$  and  $\eta$  to explicitly name those corresponding to unary connectives/operations (type  $\sigma \Rightarrow \sigma$ ).

## 1.1 Encoding Boolean operations

We start with an ordered algebra,

**abbreviation**  $sequ::\sigma\Rightarrow\sigma\Rightarrow bool$  (**infixr**  $\approx$  45) **where**  $A \approx B \equiv \forall w. (A w) \longleftrightarrow (B w)$

**abbreviation**  $subs::\sigma\Rightarrow\sigma\Rightarrow bool$  (**infixr**  $\preceq$  45) **where**  $A \preceq B \equiv \forall w. (A w) \longrightarrow (B w)$

**abbreviation**  $sup::\sigma\Rightarrow\sigma\Rightarrow bool$  (**infixr**  $\succeq$  45) **where**  $B \succeq A \equiv A \preceq B$

define meet and join by reusing HOL metalogical conjunction and disjunction,

**definition**  $meet::\sigma\Rightarrow\sigma\Rightarrow\sigma$  (**infixr**  $\wedge$  54) **where**  $A \wedge B \equiv \lambda w. (A w) \wedge (B w)$

**definition**  $join::\sigma\Rightarrow\sigma\Rightarrow\sigma$  (**infixr**  $\vee$  53) **where**  $A \vee B \equiv \lambda w. (A w) \vee (B w)$

and introduce further operations to obtain a Boolean 'algebra of propositions'.

**definition**  $top::\sigma$  ( $\top$ ) **where**  $\top \equiv \lambda w. True$

**definition**  $bottom::\sigma$  ( $\perp$ ) **where**  $\perp \equiv \lambda w. False$

**definition**  $impl::\sigma\Rightarrow\sigma\Rightarrow\sigma$  (**infixr**  $\rightarrow$  51) **where**  $A \rightarrow B \equiv \lambda w. (A w) \longrightarrow (B w)$

**definition**  $dimp::\sigma\Rightarrow\sigma\Rightarrow\sigma$  (**infixr**  $\leftrightarrow$  51) **where**  $A \leftrightarrow B \equiv \lambda w. (A w) \longleftrightarrow (B w)$

**definition**  $diff::\sigma\Rightarrow\sigma\Rightarrow\sigma$  (**infixr**  $\leftarrow$  51) **where**  $A \leftarrow B \equiv \lambda w. (A w) \wedge \neg(B w)$

**definition**  $compl::\sigma\Rightarrow\sigma$  ( $-$  [57]58) **where**  $-A \equiv \lambda w. \neg(A w)$

**named-theorems** *conn*

**declare**  $meet-def[conn]$   $join-def[conn]$   $top-def[conn]$   $bottom-def[conn]$

$impl-def[conn]$   $dimp-def[conn]$   $diff-def[conn]$   $compl-def[conn]$

Quite trivially, we can verify that the algebra satisfies some essential lattice properties.

**lemma**  $a \vee a \approx a$  *<proof>*

**lemma**  $a \wedge a \approx a$  *<proof>*

**lemma**  $a \preceq a \vee b$  *<proof>*

**lemma**  $a \wedge b \preceq a$  *<proof>*

**lemma**  $(a \wedge b) \vee b \approx b$  *<proof>*

**lemma**  $a \wedge (a \vee b) \approx a$  *<proof>*

**lemma**  $a \preceq c \implies b \preceq c \implies a \vee b \preceq c$  *<proof>*

**lemma**  $c \preceq a \implies c \preceq b \implies c \preceq a \wedge b$  *<proof>*

**lemma**  $a \preceq b \equiv (a \vee b) \approx b$  *<proof>*

**lemma**  $b \preceq a \equiv (a \wedge b) \approx b$  *<proof>*

**lemma**  $a \preceq c \implies b \preceq d \implies (a \vee b) \preceq (c \vee d)$  *<proof>*

**lemma**  $a \preceq c \implies b \preceq d \implies (a \wedge b) \preceq (c \wedge d)$  *<proof>*

## 1.2 Second-order operations and fixed-points

We define equality for propositional functions as follows.

**definition**  $equal-op::('t\Rightarrow\sigma)\Rightarrow('t\Rightarrow\sigma)\Rightarrow bool$  (**infix**  $\equiv$  60) **where**  $\varphi \equiv \psi \equiv \forall X. \varphi X \approx \psi X$

Moreover, we define some useful Boolean (2nd-order) operations on propositional functions,

**abbreviation**  $unionOp::('t\Rightarrow\sigma)\Rightarrow('t\Rightarrow\sigma)\Rightarrow('t\Rightarrow\sigma)$  (**infixr**  $\sqcup$  61) **where**  $\varphi \sqcup \psi \equiv \lambda X. \varphi X \vee \psi X$

**abbreviation**  $interOp::('t\Rightarrow\sigma)\Rightarrow('t\Rightarrow\sigma)\Rightarrow('t\Rightarrow\sigma)$  (**infixr**  $\sqcap$  62) **where**  $\varphi \sqcap \psi \equiv \lambda X. \varphi X \wedge \psi X$

**abbreviation**  $compOp::('t\Rightarrow\sigma)\Rightarrow('t\Rightarrow\sigma)$  ( $(-^c)$ ) **where**  $\varphi^c \equiv \lambda X. -\varphi X$

some of them explicitly targeting operations,

**definition**  $dual::(\sigma\Rightarrow\sigma)\Rightarrow(\sigma\Rightarrow\sigma)$  ( $(-^d)$ ) **where**  $\varphi^d \equiv \lambda X. -(\varphi(-X))$

and also define an useful operation (for technical purposes).

**definition**  $id::\sigma\Rightarrow\sigma$  ( $id$ ) **where**  $id A \equiv A$

We now prove some useful lemmas (some of them may help the provers in their hard work).

**lemma** *comp-symm*:  $\varphi^c = \psi \implies \varphi = \psi^c$  *<proof>*  
**lemma** *comp-invol*:  $\varphi^{cc} = \varphi$  *<proof>*  
**lemma** *dual-symm*:  $(\varphi \equiv \psi^d) \implies (\psi \equiv \varphi^d)$  *<proof>*  
**lemma** *dual-comp*:  $\varphi^{dc} = \varphi^{cd}$  *<proof>*

**lemma** *id<sup>d</sup>*  $\equiv id$  *<proof>*  
**lemma** *id<sup>c</sup>*  $\equiv compl$  *<proof>*  
**lemma**  $(A \sqcup B)^d \equiv (A^d) \sqcap (B^d)$  *<proof>*  
**lemma**  $(A \sqcup B)^c \equiv (A^c) \sqcap (B^c)$  *<proof>*  
**lemma**  $(A \sqcap B)^d \equiv (A^d) \sqcup (B^d)$  *<proof>*  
**lemma**  $(A \sqcap B)^c \equiv (A^c) \sqcup (B^c)$  *<proof>*

The notion of a fixed point is a fundamental one. We speak of propositions being fixed points of operations. For a given operation we define in the usual way a fixed-point predicate for propositions.

**abbreviation** *fixedpoint*:: $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow bool)$  (*fp*) **where** *fp*  $\varphi \equiv \lambda X. \varphi X \approx X$

**lemma** *fp-d*:  $(fp \varphi^d) X = (fp \varphi)(-X)$  *<proof>*  
**lemma** *fp-c*:  $(fp \varphi^c) X = (X \approx -(\varphi X))$  *<proof>*  
**lemma** *fp-dc*:  $(fp \varphi^{dc}) X = (X \approx \varphi(-X))$  *<proof>*

Indeed, we can 'operationalize' this predicate by defining a fixed-point operator as follows:

**abbreviation** *fixedpoint-op*:: $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $(-^{fp})$ ) **where**  $\varphi^{fp} \equiv \lambda X. (\varphi X) \leftrightarrow X$

**lemma** *ofp-c*:  $(\varphi^c)^{fp} \equiv (\varphi^{fp})^c$  *<proof>*  
**lemma** *ofp-d*:  $(\varphi^d)^{fp} \equiv (\varphi^{fp})^{dc}$  *<proof>*  
**lemma** *ofp-dc*:  $(\varphi^{dc})^{fp} \equiv (\varphi^{fp})^d$  *<proof>*  
**lemma** *ofp-decomp*:  $\varphi^{fp} \equiv (id \sqcap \varphi) \sqcup ((id \sqcup \varphi)^c)$  *<proof>*  
**lemma** *ofp-invol*:  $(\varphi^{fp})^{fp} \equiv \varphi$  *<proof>*

Fixed-point predicate and fixed-point operator are closely related.

**lemma** *fp-rel*:  $((fp \varphi) X) = (\varphi^{fp} X \approx \top)$  *<proof>*  
**lemma** *fp-d-rel*:  $((fp \varphi^d) X) = (\varphi^{fp}(-X) \approx \top)$  *<proof>*  
**lemma** *fp-c-rel*:  $((fp \varphi^c) X) = (\varphi^{fp} X \approx \perp)$  *<proof>*  
**lemma** *fp-dc-rel*:  $((fp \varphi^{dc}) X) = (\varphi^{fp}(-X) \approx \perp)$  *<proof>*

### 1.3 Equality and atomicity

We prove some facts about equality (which may help improve prover's performance).

**lemma** *eq-ext*:  $a \approx b \implies a = b$  *<proof>*  
**lemma** *eq-ext'*:  $a \equiv b \implies a = b$  *<proof>*  
**lemma** *meet-char*:  $a \preceq b \longleftrightarrow a \wedge b \approx a$  *<proof>*  
**lemma** *join-char*:  $a \preceq b \longleftrightarrow a \vee b \approx b$  *<proof>*

We can verify indeed that the algebra is atomic (in three different ways) by relying on the presence of primitive equality in HOL. A more general class of Boolean algebras could in principle be obtained in systems without primitive equality or by suitably restricting quantification over propositions (e.g. defining a topology and restricting quantifiers to open/closed sets).

**definition** *atom*  $a \equiv \neg(a \approx \perp) \wedge (\forall p. a \preceq p \vee a \preceq -p)$   
**lemma** *atomic1*:  $\forall w. \exists q. q w \wedge (\forall p. p w \longrightarrow q \preceq p)$  *<proof>*  
**lemma** *atomic2*:  $\forall w. \exists q. q w \wedge atom(q)$  *<proof>*  
**lemma** *atomic3*:  $\forall p. \neg(p \approx \perp) \longrightarrow (\exists q. atom(q) \wedge q \preceq p)$  *<proof>*

```

end
theory sse-boolean-algebra-quantification
  imports sse-boolean-algebra
begin
hide-const(open) List.list.Nil no-notation List.list.Nil ([])
hide-const(open) Relation.converse no-notation Relation.converse ((--1) [1000] 999)
nitpick-params[assms=true, user-axioms=true, show-all, expect=genuine, format=3]

```

## 1.4 Obtaining a complete Boolean Algebra

Our aim is to obtain a complete Boolean algebra which we can use to interpret quantified formulas (in the spirit of Boolean-valued models for set theory).

We start by defining infinite meet (infimum) and infinite join (supremum) operations,

**definition** *infimum*::  $(\sigma \Rightarrow \text{bool}) \Rightarrow \sigma$   $(\bigwedge -)$  **where**  $\bigwedge S \equiv \lambda w. \forall X. S X \longrightarrow X w$

**definition** *supremum*::  $(\sigma \Rightarrow \text{bool}) \Rightarrow \sigma$   $(\bigvee -)$  **where**  $\bigvee S \equiv \lambda w. \exists X. S X \wedge X w$

and show that the corresponding lattice is complete.

**abbreviation** *upper-bound*  $U S \equiv \forall X. (S X) \longrightarrow X \preceq U$

**abbreviation** *lower-bound*  $L S \equiv \forall X. (S X) \longrightarrow L \preceq X$

**abbreviation** *is-supremum*  $U S \equiv \text{upper-bound } U S \wedge (\forall X. \text{upper-bound } X S \longrightarrow U \preceq X)$

**abbreviation** *is-infimum*  $L S \equiv \text{lower-bound } L S \wedge (\forall X. \text{lower-bound } X S \longrightarrow X \preceq L)$

**lemma** *sup-char*: *is-supremum*  $\bigvee S S$   $\langle \text{proof} \rangle$

**lemma** *sup-ext*:  $\forall S. \exists X. \text{is-supremum } X S$   $\langle \text{proof} \rangle$

**lemma** *inf-char*: *is-infimum*  $\bigwedge S S$   $\langle \text{proof} \rangle$

**lemma** *inf-ext*:  $\forall S. \exists X. \text{is-infimum } X S$   $\langle \text{proof} \rangle$

We can check that being closed under supremum/infimum entails being closed under join/meet.

**abbreviation** *meet-closed*  $S \equiv \forall X Y. (S X \wedge S Y) \longrightarrow S(X \wedge Y)$

**abbreviation** *join-closed*  $S \equiv \forall X Y. (S X \wedge S Y) \longrightarrow S(X \vee Y)$

**abbreviation** *nonEmpty*  $S \equiv \exists x. S x$

**abbreviation** *contains*  $S D \equiv \forall X. D X \longrightarrow S X$

**abbreviation** *infimum-closed*  $S \equiv \forall D. \text{nonEmpty } D \wedge \text{contains } S D \longrightarrow S(\bigwedge D)$

**abbreviation** *supremum-closed*  $S \equiv \forall D. \text{nonEmpty } D \wedge \text{contains } S D \longrightarrow S(\bigvee D)$

**lemma** *inf-meet-closed*:  $\forall S. \text{infimum-closed } S \longrightarrow \text{meet-closed } S$   $\langle \text{proof} \rangle$

**lemma** *sup-join-closed*:  $\forall P. \text{supremum-closed } P \longrightarrow \text{join-closed } P$   $\langle \text{proof} \rangle$

## 1.5 Adding quantifiers (restricted and unrestricted)

We can harness HOL to define quantification over individuals of arbitrary type (using polymorphism). These (unrestricted) quantifiers take a propositional function and give a proposition.

**abbreviation** *mforall*::  $(t \Rightarrow \sigma) \Rightarrow \sigma$   $(\forall - [55]56)$  **where**  $\forall \pi \equiv \lambda w. \forall X. (\pi X) w$

**abbreviation** *mexists*::  $(t \Rightarrow \sigma) \Rightarrow \sigma$   $(\exists - [55]56)$  **where**  $\exists \pi \equiv \lambda w. \exists X. (\pi X) w$

To improve readability, we introduce for them an useful binder notation.

**abbreviation** *mforallB* (**binder** $\forall [55]56$ ) **where**  $\forall X. \pi X \equiv \forall \pi$

**abbreviation** *mexistsB* (**binder** $\exists [55]56$ ) **where**  $\exists X. \pi X \equiv \exists \pi$

Moreover, we define restricted quantifiers which take a 'functional domain' as additional parameter. The latter is a propositional function that maps each element 'e' to the proposition 'e exists'.

**abbreviation** *mforall-restr*::( $'t \Rightarrow \sigma \Rightarrow ('t \Rightarrow \sigma) \Rightarrow \sigma$ ) ( $\forall^R(-)$ ) **where**  $\forall^R(\delta)\pi \equiv \lambda w. \forall X. (\delta X) w \longrightarrow (\pi X) w$

**abbreviation** *mexists-restr*::( $'t \Rightarrow \sigma \Rightarrow ('t \Rightarrow \sigma) \Rightarrow \sigma$ ) ( $\exists^R(-)$ ) **where**  $\exists^R(\delta)\pi \equiv \lambda w. \exists X. (\delta X) w \wedge (\pi X) w$

## 1.6 Relating quantifiers with further operators

The following 'type-lifting' function is useful for converting sets into 'rigid' propositional functions.

**abbreviation** *lift-conv*::( $'t \Rightarrow bool \Rightarrow ('t \Rightarrow \sigma)$ ) ( $\Downarrow$ ) **where**  $\Downarrow S \equiv \lambda X. \lambda w. S X$

We introduce an useful operator: the range of a propositional function (resp. restricted over a domain),

**definition** *pfunRange*::( $'t \Rightarrow \sigma \Rightarrow (\sigma \Rightarrow bool)$ ) ( $Ra(-)$ ) **where**  $Ra(\pi) \equiv \lambda Y. \exists x. (\pi x) = Y$

**definition** *pfunRange-restr*::( $'t \Rightarrow \sigma \Rightarrow ('t \Rightarrow bool) \Rightarrow (\sigma \Rightarrow bool)$ ) ( $Ra[\pi|D]$ ) **where**  $Ra[\pi|D] \equiv \lambda Y. \exists x. (D x) \wedge (\pi x) = Y$

and check that taking infinite joins/meets (suprema/infima) over the range of a propositional function can be equivalently codified by using quantifiers. This is a quite useful simplifying relationship.

**lemma** *Ra-all*:  $\bigwedge Ra(\pi) = \forall \pi \langle proof \rangle$

**lemma** *Ra-ex*:  $\bigvee Ra(\pi) = \exists \pi \langle proof \rangle$

**lemma** *Ra-restr-all*:  $\bigwedge Ra[\pi|D] = \forall^R \Downarrow D \pi \langle proof \rangle$

**lemma** *Ra-restr-ex*:  $\bigvee Ra[\pi|D] = \exists^R \Downarrow D \pi \langle proof \rangle$

We further introduce the positive (negative) restriction of a propositional function wrt. a domain,

**abbreviation** *pfunRestr-pos*::( $'t \Rightarrow \sigma \Rightarrow ('t \Rightarrow \sigma) \Rightarrow ('t \Rightarrow \sigma)$ ) ( $[\_|\_]^P$ ) **where**  $[\pi|\delta]^P \equiv \lambda X. \lambda w. (\delta X) w \longrightarrow (\pi X) w$

**abbreviation** *pfunRestr-neg*::( $'t \Rightarrow \sigma \Rightarrow ('t \Rightarrow \sigma) \Rightarrow ('t \Rightarrow \sigma)$ ) ( $[\_|\_]^N$ ) **where**  $[\pi|\delta]^N \equiv \lambda X. \lambda w. (\delta X) w \wedge (\pi X) w$

and check that some additional simplifying relationships obtain.

**lemma** *all-restr*:  $\forall^R(\delta)\pi = \forall [\pi|\delta]^P \langle proof \rangle$

**lemma** *ex-restr*:  $\exists^R(\delta)\pi = \exists [\pi|\delta]^N \langle proof \rangle$

**lemma** *Ra-all-restr*:  $\bigwedge Ra[\pi|D] = \forall [\pi|\Downarrow D]^P \langle proof \rangle$

**lemma** *Ra-ex-restr*:  $\bigvee Ra[\pi|D] = \exists [\pi|\Downarrow D]^N \langle proof \rangle$

Observe that using these operators has the advantage of allowing for binder notation,

**lemma**  $\forall X. [\pi|\delta]^P X = \forall [\pi|\delta]^P \langle proof \rangle$

**lemma**  $\exists X. [\pi|\delta]^N X = \exists [\pi|\delta]^N \langle proof \rangle$

noting that extra care should be taken when working with complements or negations; always remember to switch P/N (positive/negative restriction) accordingly.

**lemma**  $\forall^R(\delta)\pi = \forall X. [\pi|\delta]^P X \langle proof \rangle$

**lemma**  $\forall^R(\delta)\pi^c = \forall X. -[\pi|\delta]^N X \langle proof \rangle$

**lemma**  $\exists^R(\delta)\pi = \exists X. [\pi|\delta]^N X \langle proof \rangle$

**lemma**  $\exists^R(\delta)\pi^c = \exists X. -[\pi|\delta]^P X \langle proof \rangle$

The previous definitions allow us to nicely characterize the interaction between function composition and (restricted) quantification:

**lemma** *Ra-all-comp1*:  $\forall (\pi \circ \gamma) = \forall [\pi|\Downarrow Ra \gamma]^P \langle proof \rangle$

**lemma** *Ra-all-comp2*:  $\forall (\pi \circ \gamma) = \forall^R (\text{Ra } \gamma) \pi$  *<proof>*  
**lemma** *Ra-ex-comp1*:  $\exists (\pi \circ \gamma) = \exists [\pi | (\text{Ra } \gamma)]^N$  *<proof>*  
**lemma** *Ra-ex-comp2*:  $\exists (\pi \circ \gamma) = \exists^R (\text{Ra } \gamma) \pi$  *<proof>*

This useful operator returns for a given domain of propositions the domain of their complements:

**definition** *dom-compl*:: $(\sigma \Rightarrow \text{bool}) \Rightarrow (\sigma \Rightarrow \text{bool}) ((-^1))$  **where**  $D^{-1} \equiv \lambda X. \exists Y. (D Y) \wedge (X = -Y)$

**lemma** *dom-compl-def2*:  $D^{-1} = (\lambda X. D(-X))$  *<proof>*

**lemma** *dom-compl-invol*:  $D = (D^{-1})^{-1}$  *<proof>*

We can now check an infinite variant of the De Morgan laws,

**lemma** *iDM-a*:  $-(\bigwedge S) = \bigvee S^{-1}$  *<proof>*

**lemma** *iDM-b*:  $-(\bigvee S) = \bigwedge S^{-1}$  *<proof>*

and some useful dualities regarding the range of propositional functions (restricted wrt. a domain).

**lemma** *Ra-compl*:  $\text{Ra}[\pi^c | D] = \text{Ra}[\pi | D]^{-1}$  *<proof>*

**lemma** *Ra-dual1*:  $\text{Ra}[\pi^d | D] = \text{Ra}[\pi | D^{-1}]^{-1}$  *<proof>*

**lemma** *Ra-dual2*:  $\text{Ra}[\pi^d | D] = \text{Ra}[\pi^c | D^{-1}]$  *<proof>*

**lemma** *Ra-dual3*:  $\text{Ra}[\pi^d | D]^{-1} = \text{Ra}[\pi | D^{-1}]$  *<proof>*

**lemma** *Ra-dual4*:  $\text{Ra}[\pi^d | D^{-1}] = \text{Ra}[\pi | D]^{-1}$  *<proof>*

Finally, we check some facts concerning duality for quantifiers.

**lemma**  $\exists \pi^c = -(\forall \pi)$  *<proof>*

**lemma**  $\forall \pi^c = -(\exists \pi)$  *<proof>*

**lemma**  $\exists X. -\pi X = -(\forall X. \pi X)$  *<proof>*

**lemma**  $\forall X. -\pi X = -(\exists X. \pi X)$  *<proof>*

**lemma**  $\exists^R (\delta) \pi^c = -(\forall^R (\delta) \pi)$  *<proof>*

**lemma**  $\forall^R (\delta) \pi^c = -(\exists^R (\delta) \pi)$  *<proof>*

**lemma**  $\exists X. -[\pi | \delta]^P X = -(\forall X. [\pi | \delta]^P X)$  *<proof>*

**lemma**  $\forall X. -[\pi | \delta]^P X = -(\exists X. [\pi | \delta]^P X)$  *<proof>*

**lemma**  $\exists X. -[\pi | \delta]^N X = -(\forall X. [\pi | \delta]^N X)$  *<proof>*

**lemma**  $\forall X. -[\pi | \delta]^N X = -(\exists X. [\pi | \delta]^N X)$  *<proof>*

Warning: Do not switch P and N when passing to the dual form.

**lemma**  $\forall X. [\pi | \delta]^P X = -(\exists X. -[\pi | \delta]^N X)$  **nitpick** *<proof>*

**lemma**  $\forall X. [\pi | \delta]^P X = -(\exists X. -[\pi | \delta]^P X)$  *<proof>*

**end**

**theory** *sse-operation-positive*

**imports** *sse-boolean-algebra*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 2 Positive semantic conditions for operations

We define and interrelate some useful conditions on propositional functions which do not involve negative-like properties (hence 'positive'). We focus on propositional functions which correspond to unary connectives of the algebra (with type  $\sigma \Rightarrow \sigma$ ). We call such propositional functions 'operations'.



## 2.1 Definitions (finitary case)

Monotonicity (MONO).

**definition** *MONO*  $\varphi \equiv \forall A B. A \preceq B \longrightarrow \varphi A \preceq \varphi B$

**lemma** *MONO-ant*: *MONO*  $\varphi \implies \forall A B C. A \preceq B \longrightarrow \varphi(B \rightarrow C) \preceq \varphi(A \rightarrow C)$  *<proof>*

**lemma** *MONO-cons*: *MONO*  $\varphi \implies \forall A B C. A \preceq B \longrightarrow \varphi(C \rightarrow A) \preceq \varphi(C \rightarrow B)$  *<proof>*

**lemma** *MONO-dual*: *MONO*  $\varphi \implies \text{MONO } \varphi^d$  *<proof>*

Extensive/expansive (EXP) and its dual (dEXP), aka. 'contractive'.

**definition** *EXP*  $\varphi \equiv \forall A. A \preceq \varphi A$

**definition** *dEXP*  $\varphi \equiv \forall A. \varphi A \preceq A$

**lemma** *EXP-dual1*: *EXP*  $\varphi \implies \text{dEXP } \varphi^d$  *<proof>*

**lemma** *EXP-dual2*: *dEXP*  $\varphi \implies \text{EXP } \varphi^d$  *<proof>*

Idempotence (IDEM).

**definition** *IDEM*  $\varphi \equiv \forall A. (\varphi A) \approx \varphi(\varphi A)$

**definition** *IDEMa*  $\varphi \equiv \forall A. (\varphi A) \preceq \varphi(\varphi A)$

**definition** *IDEMb*  $\varphi \equiv \forall A. (\varphi A) \succeq \varphi(\varphi A)$

**lemma** *IDEM-dual1*: *IDEMa*  $\varphi \implies \text{IDEMb } \varphi^d$  *<proof>*

**lemma** *IDEM-dual2*: *IDEMb*  $\varphi \implies \text{IDEMa } \varphi^d$  *<proof>*

**lemma** *IDEM-dual*: *IDEM*  $\varphi = \text{IDEM } \varphi^d$  *<proof>*

Normality (NOR) and its dual (dNOR).

**definition** *NOR*  $\varphi \equiv (\varphi \perp) \approx \perp$

**definition** *dNOR*  $\varphi \equiv (\varphi \top) \approx \top$

**lemma** *NOR-dual1*: *NOR*  $\varphi = \text{dNOR } \varphi^d$  *<proof>*

**lemma** *NOR-dual2*: *dNOR*  $\varphi = \text{NOR } \varphi^d$  *<proof>*

Distribution over meets or multiplicativity (MULT).

**definition** *MULT*  $\varphi \equiv \forall A B. \varphi(A \wedge B) \approx (\varphi A) \wedge (\varphi B)$

**definition** *MULT-a*  $\varphi \equiv \forall A B. \varphi(A \wedge B) \preceq (\varphi A) \wedge (\varphi B)$

**definition** *MULT-b*  $\varphi \equiv \forall A B. \varphi(A \wedge B) \succeq (\varphi A) \wedge (\varphi B)$

Distribution over joins or additivity (ADDI).

**definition** *ADDI*  $\varphi \equiv \forall A B. \varphi(A \vee B) \approx (\varphi A) \vee (\varphi B)$

**definition** *ADDI-a*  $\varphi \equiv \forall A B. \varphi(A \vee B) \preceq (\varphi A) \vee (\varphi B)$

**definition** *ADDI-b*  $\varphi \equiv \forall A B. \varphi(A \vee B) \succeq (\varphi A) \vee (\varphi B)$

## 2.2 Relations among conditions (finitary case)

dEXP and dNOR entail NOR.

**lemma** *dEXP*  $\varphi \implies \text{dNOR } \varphi \implies \text{NOR } \varphi$  *<proof>*

EXP and NOR entail dNOR.

**lemma** *EXP*  $\varphi \implies \text{NOR } \varphi \implies \text{dNOR } \varphi$  *<proof>*

Interestingly, EXP and its dual allow for an alternative characterization of fixed-point operators.

**lemma** *EXP-fp*: *EXP*  $\varphi \implies \varphi^{fp} \equiv (\varphi^c \sqcup id)$  *<proof>*

**lemma** *dEXP-fp*: *dEXP*  $\varphi \implies \varphi^{fp} \equiv (\varphi \sqcup compl)$  *<proof>*

MONO, MULT-a and ADDI-b are equivalent.

**lemma** *MONO-MULTa*: *MONO*  $\varphi = \text{MULT-a } \varphi$  *<proof>*

**lemma** *MONO-ADDIb*:  $MONO \varphi = ADDI\text{-}b \varphi$  *<proof>*  
**lemma** *ADDIb-MULTa*:  $ADDI\text{-}b \varphi = MULT\text{-}a \varphi$  *<proof>*

**end**

**theory** *sse-operation-positive-quantification*

**imports** *sse-operation-positive sse-boolean-algebra-quantification*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 2.3 Definitions (infinitary case)

We define and interrelate infinitary variants for some previously introduced ('positive') conditions on operations and show how they relate to quantifiers as previously defined.

Distribution over infinite meets (infima) or infinite multiplicativity (iMULT).

**definition** *iMULT*  $\varphi \equiv \forall S. \varphi(\bigwedge S) \approx \bigwedge Ra[\varphi|S]$

**definition** *iMULT-a*  $\varphi \equiv \forall S. \varphi(\bigwedge S) \preceq \bigwedge Ra[\varphi|S]$

**definition** *iMULT-b*  $\varphi \equiv \forall S. \varphi(\bigwedge S) \succeq \bigwedge Ra[\varphi|S]$

Distribution over infinite joins (suprema) or infinite additivity (iADDI).

**definition** *iADDI*  $\varphi \equiv \forall S. \varphi(\bigvee S) \approx \bigvee Ra[\varphi|S]$

**definition** *iADDI-a*  $\varphi \equiv \forall S. \varphi(\bigvee S) \preceq \bigvee Ra[\varphi|S]$

**definition** *iADDI-b*  $\varphi \equiv \forall S. \varphi(\bigvee S) \succeq \bigvee Ra[\varphi|S]$

## 2.4 Relations among conditions (infinitary case)

We start by noting that there is a duality between iADDI-a and iMULT-b.

**lemma** *iADDI-MULT-dual1*:  $iADDI\text{-}a \varphi \implies iMULT\text{-}b \varphi^d$  *<proof>*

**lemma** *iADDI-MULT-dual2*:  $iMULT\text{-}b \varphi \implies iADDI\text{-}a \varphi^d$  *<proof>*

MULT-a and iMULT-a are equivalent.

**lemma** *iMULTa-rel*:  $iMULT\text{-}a \varphi = MULT\text{-}a \varphi$  *<proof>*

ADDI-b and iADDI-b are equivalent.

**lemma** *iADDIb-rel*:  $iADDI\text{-}b \varphi = ADDI\text{-}b \varphi$  *<proof>*

Thus we have that MONO, MULT-a/iMULT-a and ADDI-b/iADDI-b are all equivalent.

**lemma** *MONO-iADDIb*:  $MONO \varphi = iADDI\text{-}b \varphi$  *<proof>*

**lemma** *MONO-iMULTa*:  $MONO \varphi = iMULT\text{-}a \varphi$  *<proof>*

**lemma** *iADDI-b-iMULTa*:  $iADDI\text{-}b \varphi = iMULT\text{-}a \varphi$  *<proof>*

**lemma** *PI-impl*:  $MONO \varphi \implies iMULT\text{-}b \varphi \implies iMULT \varphi$  *<proof>*

**lemma** *PC-iaddi*:  $MONO \varphi \implies iADDI\text{-}a \varphi \implies iADDI \varphi$  *<proof>*

Interestingly, we can show that suitable (infinitary) conditions on an operation can make the set of its fixed points closed under infinite meets/joins.

**lemma** *fp-inf-closed*:  $MONO \varphi \implies iMULT\text{-}b \varphi \implies \text{infimum-closed}(\text{fp } \varphi)$  *<proof>*

**lemma** *fp-sup-closed*:  $MONO \varphi \implies iADDI\text{-}a \varphi \implies \text{supremum-closed}(\text{fp } \varphi)$  *<proof>*

## 2.5 Exploring the Barcan formula and its converse

The converse Barcan formula follows readily from monotonicity.

**lemma** *CBarcan1*: *MONO*  $\varphi \implies \forall \pi. \varphi(\forall x. \pi x) \preceq (\forall x. \varphi(\pi x))$  *<proof>*

**lemma** *CBarcan2*: *MONO*  $\varphi \implies \forall \pi. (\exists x. \varphi(\pi x)) \preceq \varphi(\exists x. \pi x)$  *<proof>*

However, the Barcan formula requires a stronger assumption (of an infinitary character).

**lemma** *Barcan1*: *iMULT-b*  $\varphi \implies \forall \pi. (\forall x. \varphi(\pi x)) \preceq \varphi(\forall x. \pi x)$  *<proof>*

**lemma** *Barcan2*: *iADDI-a*  $\varphi \implies \forall \pi. \varphi(\exists x. \pi x) \preceq (\exists x. \varphi(\pi x))$  *<proof>*

**end**

**theory** *sse-operation-negative*

**imports** *sse-boolean-algebra*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

### 3 Negative semantic conditions for operations

We define and interrelate some conditions on operations (i.e. propositional functions of type  $\sigma \Rightarrow \sigma$ ), this time involving negative-like properties.

**named-theorems** *Defs*

#### 3.1 Definitions and interrelations (finitary case)

##### 3.1.1 Principles of excluded middle, contradiction and explosion

TND: tertium non datur, aka. law of excluded middle (resp. strong, weak, minimal).

**abbreviation** *pTND* (*TND<sup>-</sup>* -) **where**  $TND^a \eta \equiv \top \approx a \vee (\eta a)$

**abbreviation** *pTNDw* (*TNDw<sup>-</sup>* -) **where**  $TNDw^a \eta \equiv \forall b. (\eta b) \preceq a \vee (\eta a)$

**abbreviation** *pTNDm* (*TNDm<sup>-</sup>* -) **where**  $TNDm^a \eta \equiv (\eta \perp) \preceq a \vee (\eta a)$

**definition** *TND*  $\eta \equiv \forall \varphi. TND^\varphi \eta$

**definition** *TNDw*  $\eta \equiv \forall \varphi. TNDw^\varphi \eta$

**definition** *TNDm*  $\eta \equiv \forall \varphi. TNDm^\varphi \eta$

**declare** *TND-def*[*Defs*] *TNDw-def*[*Defs*] *TNDm-def*[*Defs*]

Explore some (non)entailment relations:

**lemma** *TND*  $\eta \implies TNDw \eta$  *<proof>*

**lemma** *TNDw*  $\eta \implies TND \eta$  **nitpick** *<proof>*

**lemma** *TNDw*  $\eta \implies TNDm \eta$  *<proof>*

**lemma** *TNDm*  $\eta \implies TNDw \eta$  **nitpick** *<proof>*

ECQ: ex contradictione (sequitur) quodlibet (resp: strong, weak, minimal).

**abbreviation** *pECQ* (*ECQ<sup>-</sup>* -) **where**  $ECQ^a \eta \equiv a \wedge (\eta a) \approx \perp$

**abbreviation** *pECQw* (*ECQw<sup>-</sup>* -) **where**  $ECQw^a \eta \equiv \forall b. a \wedge (\eta a) \preceq (\eta b)$

**abbreviation** *pECQm* (*ECQm<sup>-</sup>* -) **where**  $ECQm^a \eta \equiv a \wedge (\eta a) \preceq (\eta \top)$

**definition** *ECQ*  $\eta \equiv \forall a. ECQ^a \eta$

**definition** *ECQw*  $\eta \equiv \forall a. ECQw^a \eta$

**definition** *ECQm*  $\eta \equiv \forall a. ECQm^a \eta$

**declare** *ECQ-def*[*Defs*] *ECQw-def*[*Defs*] *ECQm-def*[*Defs*]

Explore some (non)entailment relations:

**lemma** *ECQ*  $\eta \implies ECQw \eta$  *<proof>*

**lemma** *ECQw*  $\eta \implies ECQ \eta$  **nitpick** *<proof>*

**lemma** *ECQw*  $\eta \implies ECQm \eta$  *<proof>*

**lemma** *ECQm*  $\eta \implies ECQw \eta$  **nitpick** *<proof>*

LNK: law of non-contradiction.

**abbreviation**  $pLNC$  ( $LNC^-$  -) **where**  $LNC^a \eta \equiv \eta(a \wedge \eta a) \approx \top$

**definition**  $LNC \eta \equiv \forall a. LNC^a \eta$

**declare**  $LNC-def[Defs]$

ECQ and LNC are in general independent.

**lemma**  $ECQ \eta \implies LNC \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $LNC \eta \implies ECQm \eta$  **nitpick**  $\langle proof \rangle$

### 3.1.2 Contraposition rules

CoP: contraposition (global/rule variants, resp. weak, strong var. 1, strong var. 2, strong var. 3).

**abbreviation**  $pCoPw$  ( $CoPw^{--}$  -) **where**  $CoPw^{ab} \eta \equiv a \preceq b \longrightarrow (\eta b) \preceq (\eta a)$

**abbreviation**  $pCoP1$  ( $CoP1^{--}$  -) **where**  $CoP1^{ab} \eta \equiv a \preceq (\eta b) \longrightarrow b \preceq (\eta a)$

**abbreviation**  $pCoP2$  ( $CoP2^{--}$  -) **where**  $CoP2^{ab} \eta \equiv (\eta a) \preceq b \longrightarrow (\eta b) \preceq a$

**abbreviation**  $pCoP3$  ( $CoP3^{--}$  -) **where**  $CoP3^{ab} \eta \equiv (\eta a) \preceq (\eta b) \longrightarrow b \preceq a$

**definition**  $CoPw \eta \equiv \forall a b. CoPw^{ab} \eta$

**definition**  $CoP1 \eta \equiv \forall a b. CoP1^{ab} \eta$

**definition**  $CoP1' \eta \equiv \forall a b. a \preceq (\eta b) \longleftrightarrow b \preceq (\eta a)$

**definition**  $CoP2 \eta \equiv \forall a b. CoP2^{ab} \eta$

**definition**  $CoP2' \eta \equiv \forall a b. (\eta a) \preceq b \longleftrightarrow (\eta b) \preceq a$

**definition**  $CoP3 \eta \equiv \forall a b. CoP3^{ab} \eta$

**declare**  $CoPw-def[Defs]$   $CoP1-def[Defs]$   $CoP1'-def[Defs]$

$CoP2-def[Defs]$   $CoP2'-def[Defs]$   $CoP3-def[Defs]$

**lemma**  $CoP1-defs-rel: CoP1 \eta = CoP1' \eta$   $\langle proof \rangle$

**lemma**  $CoP2-defs-rel: CoP2 \eta = CoP2' \eta$   $\langle proof \rangle$

Explore some (non)entailment relations:

**lemma**  $CoP1 \eta \implies CoPw \eta$   $\langle proof \rangle$

**lemma**  $CoPw \eta \implies CoP1 \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $CoP2 \eta \implies CoPw \eta$   $\langle proof \rangle$

**lemma**  $CoPw \eta \implies CoP2 \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $CoP3 \eta \implies CoPw \eta$   $\langle proof \rangle$

**lemma**  $CoPw \eta \implies CoP3 \eta$  **nitpick**  $\langle proof \rangle$

All three strong variants are pairwise independent. However, CoP3 follows from CoP1 plus CoP2.

**lemma**  $CoP123: CoP1 \eta \implies CoP2 \eta \implies CoP3 \eta$   $\langle proof \rangle$

Taking all CoP together still leaves room for a boldly paraconsistent resp. paracomplete logic.

**lemma**  $CoP1 \eta \implies CoP2 \eta \implies ECQm \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $CoP1 \eta \implies CoP2 \eta \implies TNDm \eta$  **nitpick**  $\langle proof \rangle$

### 3.1.3 Modus tollens rules

MT: modus (tollendo) tollens (global/rule variants).

**abbreviation**  $pMT0$  ( $MT0^{--}$  -) **where**  $MT0^{ab} \eta \equiv a \preceq b \wedge (\eta b) \approx \top \longrightarrow (\eta a) \approx \top$

**abbreviation**  $pMT1$  ( $MT1^{--}$  -) **where**  $MT1^{ab} \eta \equiv a \preceq (\eta b) \wedge b \approx \top \longrightarrow (\eta a) \approx \top$

**abbreviation**  $pMT2$  ( $MT2^{--}$  -) **where**  $MT2^{ab} \eta \equiv (\eta a) \preceq b \wedge (\eta b) \approx \top \longrightarrow a \approx \top$

**abbreviation**  $pMT3$  ( $MT3^{--}$  -) **where**  $MT3^{ab} \eta \equiv (\eta a) \preceq (\eta b) \wedge b \approx \top \longrightarrow a \approx \top$

**definition**  $MT0 \eta \equiv \forall a b. MT0^{ab} \eta$

**definition**  $MT1 \eta \equiv \forall a b. MT1^{ab} \eta$   
**definition**  $MT2 \eta \equiv \forall a b. MT2^{ab} \eta$   
**definition**  $MT3 \eta \equiv \forall a b. MT3^{ab} \eta$   
**declare**  $MT0-def[Defs]$   $MT1-def[Defs]$   $MT2-def[Defs]$   $MT3-def[Defs]$

Again, all MT variants are pairwise independent. We explore some (non)entailment relations:

**lemma**  $CoPw \eta \implies MT0 \eta$  *<proof>*  
**lemma**  $CoP1 \eta \implies MT1 \eta$  *<proof>*  
**lemma**  $CoP2 \eta \implies MT2 \eta$  *<proof>*  
**lemma**  $CoP3 \eta \implies MT3 \eta$  *<proof>*  
**lemma**  $MT0 \eta \implies MT1 \eta \implies MT2 \eta \implies MT3 \eta \implies CoPw \eta$  **nitpick** *<proof>*  
**lemma**  $MT0 \eta \implies MT1 \eta \implies MT2 \eta \implies MT3 \eta \implies ECQm \eta$  **nitpick** *<proof>*  
**lemma**  $MT0 \eta \implies MT1 \eta \implies MT2 \eta \implies MT3 \eta \implies TNDm \eta$  **nitpick** *<proof>*  
**lemma**  $MT123: MT1 \eta \implies MT2 \eta \implies MT3 \eta$  *<proof>*

### 3.1.4 Double negation introduction and elimination

DNI/DNE: double negation introduction/elimination (as axioms).

**abbreviation**  $pDNI$  ( $DNI^-$  -) **where**  $DNI^a \eta \equiv a \preceq \eta$  ( $\eta a$ )  
**abbreviation**  $pDNE$  ( $DNE^-$  -) **where**  $DNE^a \eta \equiv \eta$  ( $\eta a$ )  $\preceq a$   
**definition**  $DNI \eta \equiv \forall a. DNI^a \eta$   
**definition**  $DNE \eta \equiv \forall a. DNE^a \eta$   
**declare**  $DNI-def[Defs]$   $DNE-def[Defs]$

CoP1 (resp. CoP2) can alternatively be defined as CoPw plus DNI (resp. DNE).

**lemma**  $DNI \eta \implies CoP1 \eta$  **nitpick** *<proof>*  
**lemma**  $CoP1-def2: CoP1 \eta = (CoPw \eta \wedge DNI \eta)$  *<proof>*  
**lemma**  $DNE \eta \implies CoP2 \eta$  **nitpick** *<proof>*  
**lemma**  $CoP2-def2: CoP2 \eta = (CoPw \eta \wedge DNE \eta)$  *<proof>*

Explore some non-entailment relations:

**lemma**  $DNI \eta \implies DNE \eta \implies CoPw \eta$  **nitpick** *<proof>*  
**lemma**  $DNI \eta \implies DNE \eta \implies TNDm \eta$  **nitpick** *<proof>*  
**lemma**  $DNI \eta \implies DNE \eta \implies ECQm \eta$  **nitpick** *<proof>*  
**lemma**  $DNI \eta \implies DNE \eta \implies MT0 \eta$  **nitpick** *<proof>*  
**lemma**  $DNI \eta \implies DNE \eta \implies MT1 \eta$  **nitpick** *<proof>*  
**lemma**  $DNI \eta \implies DNE \eta \implies MT2 \eta$  **nitpick** *<proof>*  
**lemma**  $DNI \eta \implies DNE \eta \implies MT3 \eta$  **nitpick** *<proof>*

DNI/DNE: double negation introduction/elimination (as rules).

**abbreviation**  $prDNI$  ( $rDNI^-$  -) **where**  $rDNI^a \eta \equiv a \approx \top \longrightarrow \eta$  ( $\eta a$ )  $\approx \top$   
**abbreviation**  $prDNE$  ( $rDNE^-$  -) **where**  $rDNE^a \eta \equiv \eta$  ( $\eta a$ )  $\approx \top \longrightarrow a \approx \top$   
**definition**  $rDNI \eta \equiv \forall a. rDNI^a \eta$   
**definition**  $rDNE \eta \equiv \forall a. rDNE^a \eta$   
**declare**  $rDNI-def[Defs]$   $rDNE-def[Defs]$

The rule variants are strictly weaker than the axiom variants,

**lemma**  $DNI \eta \implies rDNI \eta$  *<proof>*  
**lemma**  $rDNI \eta \implies DNI \eta$  **nitpick** *<proof>*  
**lemma**  $DNE \eta \implies rDNE \eta$  *<proof>*  
**lemma**  $rDNE \eta \implies DNE \eta$  **nitpick** *<proof>*

and follow already from modus tollens.

**lemma**  $MT1-rDNI: MT1 \eta \implies rDNI \eta$  *<proof>*  
**lemma**  $MT2-rDNE: MT2 \eta \implies rDNE \eta$  *<proof>*

### 3.1.5 Normality and its dual

n(D)Nor: negative (dual) 'normality'.

**definition**  $nNor \eta \equiv (\eta \perp) \approx \top$

**definition**  $nDNor \eta \equiv (\eta \top) \approx \perp$

**declare**  $nNor\text{-def}[Defs]$   $nDNor\text{-def}[Defs]$

nNor (resp. nDNor) is entailed by CoP1 (resp. CoP2).

**lemma**  $CoP1\text{-Nor}: CoP1 \eta \implies nNor \eta$  *<proof>*

**lemma**  $CoP2\text{-DNor}: CoP2 \eta \implies nDNor \eta$  *<proof>*

**lemma**  $DNI \eta \implies nNor \eta$  **nitpick** *<proof>*

**lemma**  $DNE \eta \implies nDNor \eta$  **nitpick** *<proof>*

nNor and nDNor together entail the rule variant of DNI (rDNI).

**lemma**  $nDNor\text{-rDNI}: nNor \eta \implies nDNor \eta \implies rDNI \eta$  *<proof>*

**lemma**  $nNor \eta \implies nDNor \eta \implies rDNE \eta$  **nitpick** *<proof>*

### 3.1.6 De Morgan laws

DM: De Morgan laws.

**abbreviation**  $pDM1 (DM1^{--} -)$  **where**  $DM1^{ab} \eta \equiv \eta(a \vee b) \preceq (\eta a) \wedge (\eta b)$

**abbreviation**  $pDM2 (DM2^{--} -)$  **where**  $DM2^{ab} \eta \equiv (\eta a) \vee (\eta b) \preceq \eta(a \wedge b)$

**abbreviation**  $pDM3 (DM3^{--} -)$  **where**  $DM3^{ab} \eta \equiv \eta(a \wedge b) \preceq (\eta a) \vee (\eta b)$

**abbreviation**  $pDM4 (DM4^{--} -)$  **where**  $DM4^{ab} \eta \equiv (\eta a) \wedge (\eta b) \preceq \eta(a \vee b)$

**definition**  $DM1 \eta \equiv \forall a b. DM1^{ab} \eta$

**definition**  $DM2 \eta \equiv \forall a b. DM2^{ab} \eta$

**definition**  $DM3 \eta \equiv \forall a b. DM3^{ab} \eta$

**definition**  $DM4 \eta \equiv \forall a b. DM4^{ab} \eta$

**declare**  $DM1\text{-def}[Defs]$   $DM2\text{-def}[Defs]$   $DM3\text{-def}[Defs]$   $DM4\text{-def}[Defs]$

CoPw, DM1 and DM2 are indeed equivalent.

**lemma**  $DM1\text{-CoPw}: DM1 \eta = CoPw \eta$  *<proof>*

**lemma**  $DM2\text{-CoPw}: DM2 \eta = CoPw \eta$  *<proof>*

**lemma**  $DM12: DM1 \eta = DM2 \eta$  *<proof>*

DM3 (resp. DM4) are entailed by CoPw together with DNE (resp. DNI).

**lemma**  $CoPw\text{-DNE-DM3}: CoPw \eta \implies DNE \eta \implies DM3 \eta$  *<proof>*

**lemma**  $CoPw\text{-DNI-DM4}: CoPw \eta \implies DNI \eta \implies DM4 \eta$  *<proof>*

From this follows that DM3 (resp. DM4) is entailed by CoP2 (resp. CoP1).

**lemma**  $CoP2\text{-DM3}: CoP2 \eta \implies DM3 \eta$  *<proof>*

**lemma**  $CoP1\text{-DM4}: CoP1 \eta \implies DM4 \eta$  *<proof>*

Explore some non-entailment relations:

**lemma**  $CoPw \eta \implies DM3 \eta \implies DM4 \eta \implies nNor \eta \implies nDNor \eta \implies DNI \eta$  **nitpick** *<proof>*

**lemma**  $CoPw \eta \implies DM3 \eta \implies DM4 \eta \implies nNor \eta \implies nDNor \eta \implies DNE \eta$  **nitpick** *<proof>*

**lemma**  $CoPw \eta \implies DM3 \eta \implies DM4 \eta \implies DNI \eta \implies DNE \eta \implies ECQm \eta$  **nitpick** *<proof>*

**lemma**  $CoPw \eta \implies DM3 \eta \implies DM4 \eta \implies DNI \eta \implies DNE \eta \implies TNDm \eta$  **nitpick** *<proof>*

### 3.1.7 Contextual (strong) contraposition rule

XCoP: contextual contraposition (global/rule variant).

**abbreviation**  $pXCoP$  ( $XCoP^{--}$  -) **where**  $XCoP^{ab} \eta \equiv \forall c. c \wedge a \preceq b \longrightarrow c \wedge (\eta b) \preceq (\eta a)$   
**definition**  $XCoP \eta \equiv \forall a b. XCoP^{ab} \eta$   
**declare**  $XCoP-def[Defs]$

XCoP can alternatively be defined as ECQw plus TNDw.

**lemma**  $XCoP-def2$ :  $XCoP \eta = (ECQw \eta \wedge TNDw \eta)$   $\langle proof \rangle$

Explore some (non)entailment relations:

**lemma**  $XCoP \eta \implies ECQ \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies TND \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP-CoPw$ :  $XCoP \eta \implies CoPw \eta$   $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies CoP1 \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies CoP2 \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies CoP3 \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $CoP1 \eta \wedge CoP2 \eta \implies XCoP \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies nNor \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies nDNor \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies rDNI \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP \eta \implies rDNE \eta$  **nitpick**  $\langle proof \rangle$   
**lemma**  $XCoP-DM3$ :  $XCoP \eta \implies DM3 \eta$   $\langle proof \rangle$   
**lemma**  $XCoP-DM4$ :  $XCoP \eta \implies DM4 \eta$   $\langle proof \rangle$

### 3.1.8 Local contraposition axioms

Observe that the definitions below take implication as an additional parameter:  $\iota$ .

lCoP: contraposition (local/axiom variants).

**abbreviation**  $plCoPw$  ( $lCoPw^{--}$  -) **where**  $lCoPw^{ab} \iota \eta \equiv (\iota a b :: \sigma) \preceq (\iota (\eta b) (\eta a))$   
**abbreviation**  $plCoP1$  ( $lCoP1^{--}$  -) **where**  $lCoP1^{ab} \iota \eta \equiv (\iota a (\eta b :: \sigma)) \preceq (\iota b (\eta a))$   
**abbreviation**  $plCoP2$  ( $lCoP2^{--}$  -) **where**  $lCoP2^{ab} \iota \eta \equiv (\iota (\eta a) b :: \sigma) \preceq (\iota (\eta b) a)$   
**abbreviation**  $plCoP3$  ( $lCoP3^{--}$  -) **where**  $lCoP3^{ab} \iota \eta \equiv (\iota (\eta a) (\eta b :: \sigma)) \preceq (\iota b a)$   
**definition**  $lCoPw \iota \eta \equiv \forall a b. lCoPw^{ab} \iota \eta$   
**definition**  $lCoP1 \iota \eta \equiv \forall a b. lCoP1^{ab} \iota \eta$   
**definition**  $lCoP1' \iota \eta \equiv \forall a b. (\iota a (\eta b)) \approx (\iota b (\eta a))$   
**definition**  $lCoP2 \iota \eta \equiv \forall a b. lCoP2^{ab} \iota \eta$   
**definition**  $lCoP2' \iota \eta \equiv \forall a b. (\iota (\eta a) b) \approx (\iota (\eta b) a)$   
**definition**  $lCoP3 \iota \eta \equiv \forall a b. lCoP3^{ab} \iota \eta$   
**declare**  $lCoPw-def[Defs]$   $lCoP1-def[Defs]$   $lCoP1'-def[Defs]$   
 $lCoP2-def[Defs]$   $lCoP2'-def[Defs]$   $lCoP3-def[Defs]$

**lemma**  $lCoP1-defs-rel$ :  $lCoP1 \iota \eta = lCoP1' \iota \eta$   $\langle proof \rangle$

**lemma**  $lCoP2-defs-rel$ :  $lCoP2 \iota \eta = lCoP2' \iota \eta$   $\langle proof \rangle$

All local contraposition variants are in general independent from each other. However if we take classical implication we can verify some relationships.

**lemma**  $lCoP1-def2$ :  $lCoP1(\rightarrow) \eta = (lCoPw(\rightarrow) \eta \wedge DNI \eta)$   $\langle proof \rangle$

**lemma**  $lCoP2-def2$ :  $lCoP2(\rightarrow) \eta = (lCoPw(\rightarrow) \eta \wedge DNE \eta)$   $\langle proof \rangle$

**lemma**  $lCoP1(\rightarrow) \eta \implies lCoPw(\rightarrow) \eta$   $\langle proof \rangle$

**lemma**  $lCoPw(\rightarrow) \eta \implies lCoP1(\rightarrow) \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $lCoP2(\rightarrow) \eta \implies lCoPw(\rightarrow) \eta$   $\langle proof \rangle$

**lemma**  $lCoPw(\rightarrow) \eta \implies lCoP2(\rightarrow) \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $lCoP3(\rightarrow) \eta \implies lCoPw(\rightarrow) \eta$   $\langle proof \rangle$

**lemma**  $lCoPw(\rightarrow) \eta \implies lCoP3(\rightarrow) \eta$  **nitpick**  $\langle proof \rangle$

**lemma**  $lCoP123$ :  $lCoP1(\rightarrow) \eta \wedge lCoP2(\rightarrow) \eta \implies lCoP3(\rightarrow) \eta$  *<proof>*

Local variants imply global ones as expected.

**lemma**  $lCoPw(\rightarrow) \eta \implies CoPw \eta$  *<proof>*

**lemma**  $lCoP1(\rightarrow) \eta \implies CoP1 \eta$  *<proof>*

**lemma**  $lCoP2(\rightarrow) \eta \implies CoP2 \eta$  *<proof>*

**lemma**  $lCoP3(\rightarrow) \eta \implies CoP3 \eta$  *<proof>*

Explore some (non)entailment relations.

**lemma**  $lCoPw-XCoP$ :  $lCoPw(\rightarrow) \eta = XCoP \eta$  *<proof>*

**lemma**  $lCoP1-TND$ :  $lCoP1(\rightarrow) \eta \implies TND \eta$  *<proof>*

**lemma**  $TND \eta \implies lCoP1(\rightarrow) \eta$  **nitpick** *<proof>*

**lemma**  $lCoP2-ECQ$ :  $lCoP2(\rightarrow) \eta \implies ECQ \eta$  *<proof>*

**lemma**  $ECQ \eta \implies lCoP2(\rightarrow) \eta$  **nitpick** *<proof>*

### 3.1.9 Local modus tollens axioms

IMT: Modus tollens (local/axiom variants).

**abbreviation**  $plMT0$  ( $lMT0^{--}$  - -) **where**  $lMT0^{ab} \iota \eta \equiv (\iota a b::\sigma) \wedge (\eta b) \preceq (\eta a)$

**abbreviation**  $plMT1$  ( $lMT1^{--}$  - -) **where**  $lMT1^{ab} \iota \eta \equiv (\iota a (\eta b::\sigma)) \wedge b \preceq (\eta a)$

**abbreviation**  $plMT2$  ( $lMT2^{--}$  - -) **where**  $lMT2^{ab} \iota \eta \equiv (\iota (\eta a) b::\sigma) \wedge (\eta b) \preceq a$

**abbreviation**  $plMT3$  ( $lMT3^{--}$  - -) **where**  $lMT3^{ab} \iota \eta \equiv (\iota (\eta a) (\eta b::\sigma)) \wedge b \preceq a$

**definition**  $lMT0 \iota \eta \equiv \forall a b. lMT0^{ab} \iota \eta$

**definition**  $lMT1 \iota \eta \equiv \forall a b. lMT1^{ab} \iota \eta$

**definition**  $lMT2 \iota \eta \equiv \forall a b. lMT2^{ab} \iota \eta$

**definition**  $lMT3 \iota \eta \equiv \forall a b. lMT3^{ab} \iota \eta$

**declare**  $lMT0-def[Defs]$   $lMT1-def[Defs]$   $lMT2-def[Defs]$   $lMT3-def[Defs]$

All local MT variants are in general independent from each other and also from local CoP instances. However if we take classical implication we can verify that local MT and CoP are indeed equivalent.

**lemma**  $lMT0(\rightarrow) \eta = lCoPw(\rightarrow) \eta$  *<proof>*

**lemma**  $lMT1(\rightarrow) \eta = lCoP1(\rightarrow) \eta$  *<proof>*

**lemma**  $lMT2(\rightarrow) \eta = lCoP2(\rightarrow) \eta$  *<proof>*

**lemma**  $lMT3(\rightarrow) \eta = lCoP3(\rightarrow) \eta$  *<proof>*

### 3.1.10 Disjunctive syllogism

DS: disjunctive syllogism.

**abbreviation**  $pDS1$  ( $DS1^{--}$  - -) **where**  $DS1^{ab} \iota \eta \equiv (a \vee b::\sigma) \preceq (\iota (\eta a) b)$

**abbreviation**  $pDS2$  ( $DS2^{--}$  - -) **where**  $DS2^{ab} \iota \eta \equiv (\iota (\eta a) b::\sigma) \preceq (a \vee b)$

**abbreviation**  $pDS3$  ( $DS3^{--}$  - -) **where**  $DS3^{ab} \iota \eta \equiv ((\eta a) \vee b::\sigma) \preceq (\iota a b)$

**abbreviation**  $pDS4$  ( $DS4^{--}$  - -) **where**  $DS4^{ab} \iota \eta \equiv (\iota a b::\sigma) \preceq ((\eta a) \vee b)$

**definition**  $DS1 \iota \eta \equiv \forall a b. DS1^{ab} \iota \eta$

**definition**  $DS2 \iota \eta \equiv \forall a b. DS2^{ab} \iota \eta$

**definition**  $DS3 \iota \eta \equiv \forall a b. DS3^{ab} \iota \eta$

**definition**  $DS4 \iota \eta \equiv \forall a b. DS4^{ab} \iota \eta$

**declare**  $DS1-def[Defs]$   $DS2-def[Defs]$   $DS3-def[Defs]$   $DS4-def[Defs]$

All DS variants are in general independent from each other. However if we take classical implication we can verify that the pairs DS1-DS3 and DS2-DS4 are indeed equivalent.

**lemma**  $DS1(\rightarrow) \eta = DS3(\rightarrow) \eta$  *<proof>*

**lemma**  $DS2(\rightarrow) \eta = DS4(\rightarrow) \eta$  *<proof>*



Explore some (non)entailment relations.

**lemma** *DS1-nDNor*:  $DS1(\rightarrow) \eta \Longrightarrow nDNor \eta$  *<proof>*  
**lemma** *DS2-nNor*:  $DS2(\rightarrow) \eta \Longrightarrow nNor \eta$  *<proof>*  
**lemma** *lCoP2-DS1*:  $lCoP2(\rightarrow) \eta \Longrightarrow DS1(\rightarrow) \eta$  *<proof>*  
**lemma** *lCoP1-DS2*:  $lCoP1(\rightarrow) \eta \Longrightarrow DS2(\rightarrow) \eta$  *<proof>*  
**lemma** *CoP2*  $\eta \Longrightarrow DS1(\rightarrow) \eta$  **nitpick** *<proof>*  
**lemma** *CoP1*  $\eta \Longrightarrow DS2(\rightarrow) \eta$  **nitpick** *<proof>*

**end**

**theory** *sse-operation-negative-quantification*

**imports** *sse-operation-negative sse-boolean-algebra-quantification*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

### 3.2 Definitions and interrelations (infinitary case)

We define and interrelate infinitary variants for some previously introduced ('negative') conditions on operations. We show how they relate to quantifiers as previously defined.

iDM: infinitary De Morgan laws.

**abbreviation** *riDM1* (*iDM1<sup>-</sup>* -) **where**  $iDM1^S \eta \equiv \eta(\bigvee S) \preceq \bigwedge Ra[\eta|S]$   
**abbreviation** *riDM2* (*iDM2<sup>-</sup>* -) **where**  $iDM2^S \eta \equiv \bigvee Ra[\eta|S] \preceq \eta(\bigwedge S)$   
**abbreviation** *riDM3* (*iDM3<sup>-</sup>* -) **where**  $iDM3^S \eta \equiv \eta(\bigwedge S) \preceq \bigvee Ra[\eta|S]$   
**abbreviation** *riDM4* (*iDM4<sup>-</sup>* -) **where**  $iDM4^S \eta \equiv \bigwedge Ra[\eta|S] \preceq \eta(\bigvee S)$   
**definition** *iDM1*  $\eta \equiv \forall S. iDM1^S \eta$   
**definition** *iDM2*  $\eta \equiv \forall S. iDM2^S \eta$   
**definition** *iDM3*  $\eta \equiv \forall S. iDM3^S \eta$   
**definition** *iDM4*  $\eta \equiv \forall S. iDM4^S \eta$   
**declare** *iDM1-def*[*Defs*] *iDM2-def*[*Defs*] *iDM3-def*[*Defs*] *iDM4-def*[*Defs*]

**lemma** *CoPw-iDM1*:  $CoPw \eta \Longrightarrow iDM1 \eta$  *<proof>*  
**lemma** *CoPw-iDM2*:  $CoPw \eta \Longrightarrow iDM2 \eta$  *<proof>*  
**lemma** *CoP2-iDM3*:  $CoP2 \eta \Longrightarrow iDM3 \eta$  *<proof>*  
**lemma** *CoP1-iDM4*:  $CoP1 \eta \Longrightarrow iDM4 \eta$  *<proof>*  
**lemma** *XCoP*  $\eta \Longrightarrow iDM3 \eta$  **nitpick** *<proof>*  
**lemma** *XCoP*  $\eta \Longrightarrow iDM4 \eta$  **nitpick** *<proof>*

DM1, DM2, iDM1, iDM2 and CoPw are equivalent.

**lemma** *iDM1-rel*:  $iDM1 \eta \Longrightarrow DM1 \eta$  *<proof>*  
**lemma** *iDM2-rel*:  $iDM2 \eta \Longrightarrow DM2 \eta$  *<proof>*  
**lemma** *DM1*  $\eta = iDM1 \eta$  *<proof>*  
**lemma** *DM2*  $\eta = iDM2 \eta$  *<proof>*  
**lemma** *iDM1*  $\eta = iDM2 \eta$  *<proof>*

iDM3/4 entail their finitary variants but not the other way round.

**lemma** *iDM3-rel*:  $iDM3 \eta \Longrightarrow DM3 \eta$  *<proof>*  
**lemma** *iDM4-rel*:  $iDM4 \eta \Longrightarrow DM4 \eta$  *<proof>*  
**lemma** *DM3*  $\eta \Longrightarrow iDM3 \eta$  **nitpick** *<proof>*  
**lemma** *DM4*  $\eta \Longrightarrow iDM4 \eta$  **nitpick** *<proof>*

Indeed the previous characterization of the infinitary De Morgan laws is fairly general and entails the traditional version employing quantifiers (though not the other way round).

The first two variants DM1/2 follow easily from DM1/2, iDM1/2 or CoPw (all of them equivalent).

**lemma** *iDM1-trad*:  $iDM1 \eta \implies \forall \pi. \eta(\exists x. \pi x) \preceq (\forall x. \eta(\pi x))$  *<proof>*

**lemma** *iDM2-trad*:  $iDM2 \eta \implies \forall \pi. (\exists x. \eta(\pi x)) \preceq \eta(\forall x. \pi x)$  *<proof>*

An analogous relationship holds for variants DM3/4, though the proof is less trivial. To see how let us first consider an intermediate version of the De Morgan laws, obtained as a particular case of the general variant above, with S as the range of a propositional function.

**abbreviation** *piDM1*  $\pi \eta \equiv \eta(\bigvee Ra(\pi)) \preceq \bigwedge Ra[\eta]Ra(\pi)$

**abbreviation** *piDM2*  $\pi \eta \equiv \bigvee Ra[\eta]Ra(\pi) \preceq \eta(\bigwedge Ra(\pi))$

**abbreviation** *piDM3*  $\pi \eta \equiv \eta(\bigwedge Ra(\pi)) \preceq \bigvee Ra[\eta]Ra(\pi)$

**abbreviation** *piDM4*  $\pi \eta \equiv \bigwedge Ra[\eta]Ra(\pi) \preceq \eta(\bigvee Ra(\pi))$

They are entailed (unidirectionally) by the general De Morgan laws.

**lemma** *iDM1*  $\eta \implies \forall \pi. piDM1 \pi \eta$  *<proof>*

**lemma** *iDM2*  $\eta \implies \forall \pi. piDM2 \pi \eta$  *<proof>*

**lemma** *iDM3*  $\eta \implies \forall \pi. piDM3 \pi \eta$  *<proof>*

**lemma** *iDM4*  $\eta \implies \forall \pi. piDM4 \pi \eta$  *<proof>*

Drawing upon the relationships shown previously we can rewrite the latter two as:

**lemma** *iDM3-aux*:  $piDM3 \pi \eta \equiv \eta(\forall \pi) \preceq \exists [\eta](\bigwedge Ra \pi)^N$  *<proof>*

**lemma** *iDM4-aux*:  $piDM4 \pi \eta \equiv \forall [\eta](\bigvee Ra \pi)^P \preceq \eta(\exists \pi)$  *<proof>*

and thus finally obtain the desired formulas.

**lemma** *iDM3-trad*:  $iDM3 \eta \implies \forall \pi. \eta(\forall x. \pi x) \preceq (\exists x. \eta(\pi x))$  *<proof>*

**lemma** *iDM4-trad*:  $iDM4 \eta \implies \forall \pi. (\forall x. \eta(\pi x)) \preceq \eta(\exists x. \pi x)$  *<proof>*

**end**

**theory** *topo-operators-basic*

**imports** *sse-operation-positive-quantification*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

**abbreviation** *implies-rl*::*bool* $\implies$ *bool* $\implies$ *bool* (**infixl**  $\longleftarrow$  25) **where**  $\varphi \longleftarrow \psi \equiv \psi \longrightarrow \varphi$

## 4 Topological operators

Below we define some conditions on algebraic operations (aka. operators) with type  $\sigma \implies \sigma$ . Those operations are aimed at extending a Boolean 'algebra of propositions' towards different generalizations of topological algebras. We divide this section into two parts. In the first we define and interrelate the topological operators of interior, closure, border and frontier. In the second we introduce the (more fundamental) notion of derivative (aka. derived set) and its related notion of (Cantorian) coherence, defining both as operators. We follow the naming conventions introduced originally by Kuratowski [8] (cf. also [9]) and Zarycki [12].

### 4.1 Interior and closure

In this section we examine the traditional notion of topological (closure, resp. interior) algebras in the spirit of McKinsey & Tarski [11], but drawing primarily from the works of Zarycki [12] and Kuratowski [8]. We also explore the less-known notions of border (cf. 'Rand' [6], 'bord' [12]) and frontier (aka. 'boundary'; cf. 'Grenze' [6], 'frontière' [12] [9]) as studied by Zarycki [12] and define corresponding operations for them.

### 4.1.1 Interior conditions

**abbreviation**  $Int-1 \varphi \equiv MULT \varphi$

**abbreviation**  $Int-1a \varphi \equiv MULT-a \varphi$

**abbreviation**  $Int-1b \varphi \equiv MULT-b \varphi$

**abbreviation**  $Int-2 \varphi \equiv dEXP \varphi$

**abbreviation**  $Int-3 \varphi \equiv dNOR \varphi$

**abbreviation**  $Int-4 \varphi \equiv IDEM \varphi$

**abbreviation**  $Int-4' \varphi \equiv IDEMa \varphi$

**abbreviation**  $Int-5 \varphi \equiv NOR \varphi$

**definition**  $Int-6 \varphi \equiv \forall A B. \varphi(A \leftarrow B) \preceq \varphi(A) \leftarrow \varphi(B)$

**definition**  $Int-7 \varphi \equiv \forall A B. \varphi(A \rightarrow B) \preceq \varphi(A) \rightarrow \varphi(B)$

**definition**  $Int-8 \varphi \equiv \forall A B. \varphi(\varphi A \vee \varphi B) \approx (\varphi A) \vee (\varphi B)$

**definition**  $Int-9 \varphi \equiv \forall A B. \varphi A \preceq B \rightarrow \varphi A \preceq \varphi B$

$\varphi$  is an interior operator ( $\mathfrak{I}(\varphi)$ ) iff it satisfies conditions 1-4 (cf. [12] and also [9]). This characterization is shown consistent by generating a non-trivial model.

**abbreviation**  $\mathfrak{I} \varphi \equiv Int-1 \varphi \wedge Int-2 \varphi \wedge Int-3 \varphi \wedge Int-4 \varphi$

**lemma**  $\mathfrak{I} \varphi$  **nitpick**[*satisfy, card w=3*] *<proof>*

We verify some properties which will become useful later (also to improve provers' performance).

**lemma**  $PI1: Int-1 \varphi = (Int-1a \varphi \wedge Int-1b \varphi)$  *<proof>*

**lemma**  $PI4: Int-2 \varphi \implies (Int-4 \varphi = Int-4' \varphi)$  *<proof>*

**lemma**  $PI5: Int-2 \varphi \implies Int-5 \varphi$  *<proof>*

**lemma**  $PI6: Int-1a \varphi \implies Int-2 \varphi \implies Int-6 \varphi$  *<proof>*

**lemma**  $PI7: Int-1 \varphi \implies Int-7 \varphi$  *<proof>*

**lemma**  $PI8: Int-1a \varphi \implies Int-2 \varphi \implies Int-4 \varphi \implies Int-8 \varphi$  *<proof>*

**lemma**  $PI9: Int-1a \varphi \implies Int-4 \varphi \implies Int-9 \varphi$  *<proof>*

### 4.1.2 Closure conditions

**abbreviation**  $Cl-1 \varphi \equiv ADDI \varphi$

**abbreviation**  $Cl-1a \varphi \equiv ADDI-a \varphi$

**abbreviation**  $Cl-1b \varphi \equiv ADDI-b \varphi$

**abbreviation**  $Cl-2 \varphi \equiv EXP \varphi$

**abbreviation**  $Cl-3 \varphi \equiv NOR \varphi$

**abbreviation**  $Cl-4 \varphi \equiv IDEM \varphi$

**abbreviation**  $Cl-4' \varphi \equiv IDEMb \varphi$

**abbreviation**  $Cl-5 \varphi \equiv dNOR \varphi$

**definition**  $Cl-6 \varphi \equiv \forall A B. (\varphi A) \leftarrow (\varphi B) \preceq \varphi(A \leftarrow B)$

**definition**  $Cl-7 \varphi \equiv \forall A B. (\varphi A) \rightarrow (\varphi B) \preceq \varphi(A \rightarrow B)$

**definition**  $Cl-8 \varphi \equiv \forall A B. \varphi(\varphi A \wedge \varphi B) \approx (\varphi A) \wedge (\varphi B)$

**definition**  $Cl-9 \varphi \equiv \forall A B. A \preceq \varphi B \rightarrow \varphi A \preceq \varphi B$

$\varphi$  is a closure operator ( $\mathfrak{C}(\varphi)$ ) iff it satisfies conditions 1-4 (cf. [8] [9]). This characterization is shown consistent by generating a non-trivial model.

**abbreviation**  $\mathfrak{C} \varphi \equiv Cl-1 \varphi \wedge Cl-2 \varphi \wedge Cl-3 \varphi \wedge Cl-4 \varphi$

**lemma**  $\mathfrak{C} \varphi$  **nitpick**[*satisfy, card w=3*] *<proof>*

We verify some properties that will become useful later.

**lemma**  $PC1: Cl-1 \varphi = (Cl-1a \varphi \wedge Cl-1b \varphi)$  *<proof>*

**lemma**  $PC4: Cl-2 \varphi \implies (Cl-4 \varphi = Cl-4' \varphi)$  *<proof>*

**lemma**  $PC5: Cl-2 \varphi \implies Cl-5 \varphi$  *<proof>*

**lemma PC6:**  $Cl-1 \varphi \implies Cl-6 \varphi$  *<proof>*  
**lemma PC7:**  $Cl-1b \varphi \implies Cl-2 \varphi \implies Cl-7 \varphi$  *<proof>*  
**lemma PC8:**  $Cl-1b \varphi \implies Cl-2 \varphi \implies Cl-4 \varphi \implies Cl-8 \varphi$  *<proof>*  
**lemma PC9:**  $Cl-1b \varphi \implies Cl-4 \varphi \implies Cl-9 \varphi$  *<proof>*

### 4.1.3 Exploring dualities

**lemma IC1-dual:**  $Int-1a \varphi = Cl-1b \varphi$  *<proof>*  
**lemma Int-1b**  $\varphi = Cl-1a \varphi$  **nitpick** *<proof>*

**lemma IC1a:**  $Int-1a \varphi \implies Cl-1b \varphi^d$  *<proof>*  
**lemma IC1b:**  $Int-1b \varphi \implies Cl-1a \varphi^d$  *<proof>*  
**lemma IC1:**  $Int-1 \varphi \implies Cl-1 \varphi^d$  *<proof>*  
**lemma IC2:**  $Int-2 \varphi \implies Cl-2 \varphi^d$  *<proof>*  
**lemma IC3:**  $Int-3 \varphi \implies Cl-3 \varphi^d$  *<proof>*  
**lemma IC4:**  $Int-4 \varphi \implies Cl-4 \varphi^d$  *<proof>*  
**lemma IC4':**  $Int-4' \varphi \implies Cl-4' \varphi^d$  *<proof>*  
**lemma IC5:**  $Int-5 \varphi \implies Cl-5 \varphi^d$  *<proof>*

**lemma CI1a:**  $Cl-1a \varphi \implies Int-1b \varphi^d$  *<proof>*  
**lemma CI1b:**  $Cl-1b \varphi \implies Int-1a \varphi^d$  *<proof>*  
**lemma CI1:**  $Cl-1 \varphi \implies Int-1 \varphi^d$  *<proof>*  
**lemma CI2:**  $Cl-2 \varphi \implies Int-2 \varphi^d$  *<proof>*  
**lemma CI3:**  $Cl-3 \varphi \implies Int-3 \varphi^d$  *<proof>*  
**lemma CI4:**  $Cl-4 \varphi \implies Int-4 \varphi^d$  *<proof>*  
**lemma CI4':**  $Cl-4' \varphi \implies Int-4' \varphi^d$  *<proof>*  
**lemma CI5:**  $Cl-5 \varphi \implies Int-5 \varphi^d$  *<proof>*

## 4.2 Frontier and border

### 4.2.1 Frontier conditions

**definition Fr-1a**  $\varphi \equiv \forall A B. (A \wedge B) \wedge \varphi(A \wedge B) \preceq (A \wedge B) \wedge (\varphi A \vee \varphi B)$   
**definition Fr-1b**  $\varphi \equiv \forall A B. (A \wedge B) \wedge \varphi(A \wedge B) \succeq (A \wedge B) \wedge (\varphi A \vee \varphi B)$   
**definition Fr-1**  $\varphi \equiv \forall A B. (A \wedge B) \wedge \varphi(A \wedge B) \approx (A \wedge B) \wedge (\varphi A \vee \varphi B)$   
**definition Fr-2**  $\varphi \equiv \forall A. \varphi A \approx \varphi(\neg A)$   
**abbreviation Fr-3**  $\varphi \equiv NOR \varphi$   
**definition Fr-4**  $\varphi \equiv \forall A. \varphi(\varphi A) \preceq (\varphi A)$

**definition Fr-5**  $\varphi \equiv \forall A. \varphi(\varphi(\varphi A)) \approx \varphi(\varphi A)$   
**definition Fr-6**  $\varphi \equiv \forall A B. A \preceq B \longrightarrow (\varphi A \preceq B \vee \varphi B)$

$\varphi$  is a topological frontier operator ( $\mathfrak{F}(\varphi)$ ) iff it satisfies conditions 1-4 (cf. [12]). This is also shown consistent by generating a non-trivial model.

**abbreviation**  $\mathfrak{F} \varphi \equiv Fr-1 \varphi \wedge Fr-2 \varphi \wedge Fr-3 \varphi \wedge Fr-4 \varphi$   
**lemma**  $\mathfrak{F} \varphi$  **nitpick**[*satisfy, card w=3*] *<proof>*

We now verify some useful properties of the frontier operator.

**lemma PF1:**  $Fr-1 \varphi = (Fr-1a \varphi \wedge Fr-1b \varphi)$  *<proof>*  
**lemma PF5:**  $Fr-1 \varphi \implies Fr-4 \varphi \implies Fr-5 \varphi$  *<proof>*  
**lemma PF6:**  $Fr-1b \varphi \implies Fr-2 \varphi \implies Fr-6 \varphi$  *<proof>*

### 4.2.2 Border conditions

**definition Br-1**  $\varphi \equiv \forall A B. \varphi(A \wedge B) \approx (A \wedge \varphi B) \vee (B \wedge \varphi A)$   
**definition Br-2**  $\varphi \equiv (\varphi \top) \approx \perp$

**definition** *Br-3*  $\varphi \equiv \forall A. \varphi(\varphi^d A) \preceq A$

**definition** *Br-4*  $\varphi \equiv \forall A B. A \preceq B \longrightarrow A \wedge (\varphi B) \preceq \varphi A$

**definition** *Br-5a*  $\varphi \equiv \forall A. \varphi(\varphi^d A) \preceq \varphi A$

**definition** *Br-5b*  $\varphi \equiv \forall A. \varphi A \preceq A$

**definition** *Br-5c*  $\varphi \equiv \forall A. A \preceq \varphi^d A$

**definition** *Br-5d*  $\varphi \equiv \forall A. \varphi^d A \preceq \varphi^d(\varphi A)$

**abbreviation** *Br-6*  $\varphi \equiv \text{IDEM } \varphi$

**abbreviation** *Br-7*  $\varphi \equiv \text{ADDI-a } \varphi$

**abbreviation** *Br-8*  $\varphi \equiv \text{MULT-b } \varphi$

**definition** *Br-9*  $\varphi \equiv \forall A B. \varphi(A \wedge B) \preceq (\varphi A) \vee (\varphi B)$

**definition** *Br-10*  $\varphi \equiv \forall A. \varphi(\neg(\varphi A) \wedge \varphi^d A) \approx \perp$

$\varphi$  is a topological border operator ( $\mathfrak{B}(\varphi)$ ) iff it satisfies conditions 1-3 (cf. [12]). This is also shown consistent.

**abbreviation**  $\mathfrak{B} \varphi \equiv \text{Br-1 } \varphi \wedge \text{Br-2 } \varphi \wedge \text{Br-3 } \varphi$

**lemma**  $\mathfrak{B} \varphi$  **nitpick**[*satisfy, card w=3*] *<proof>*

We now verify some useful properties of the border operator.

**lemma** *PB4*: *Br-1*  $\varphi \implies \text{Br-4 } \varphi$  *<proof>*

**lemma** *PB5b*: *Br-1*  $\varphi \implies \text{Br-5b } \varphi$  *<proof>*

**lemma** *PB5c*: *Br-1*  $\varphi \implies \text{Br-5c } \varphi$  *<proof>*

**lemma** *PB5a*: *Br-1*  $\varphi \implies \text{Br-3 } \varphi \implies \text{Br-5a } \varphi$  *<proof>*

**lemma** *PB5d*: *Br-1*  $\varphi \implies \text{Br-3 } \varphi \implies \text{Br-5d } \varphi$  *<proof>*

**lemma** *PB6*: *Br-1*  $\varphi \implies \text{Br-6 } \varphi$  *<proof>*

**lemma** *PB7*: *Br-1*  $\varphi \implies \text{Br-7 } \varphi$  *<proof>*

**lemma** *PB8*: *Br-1*  $\varphi \implies \text{Br-8 } \varphi$  *<proof>*

**lemma** *PB9*: *Br-1*  $\varphi \implies \text{Br-9 } \varphi$  *<proof>*

**lemma** *PB10*: *Br-1*  $\varphi \implies \text{Br-3 } \varphi \implies \text{Br-10 } \varphi$  *<proof>*

### 4.2.3 Relation with closure and interior

We define and verify some conversion operators useful to derive border and frontier operators from closure/interior operators and also between each other.

Frontier operator as derived from interior.

**definition** *Fr-int*:: $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{F}_I$ ) **where**  $\mathcal{F}_I \mathcal{I} \equiv \lambda A. \neg(\mathcal{I} A) \wedge \mathcal{I}^d A$

**lemma** *FI1*: *Int-1*  $\varphi \implies \text{Int-2 } \varphi \implies \text{Fr-1}(\mathcal{F}_I \varphi)$  *<proof>*

**lemma** *FI2*: *Fr-2*( $\mathcal{F}_I \varphi$ ) *<proof>*

**lemma** *FI3*: *Int-3*  $\varphi \implies \text{Fr-3}(\mathcal{F}_I \varphi)$  *<proof>*

**lemma** *FI4*: *Int-1a*  $\varphi \implies \text{Int-2 } \varphi \implies \text{Int-4 } \varphi \implies \text{Fr-4}(\mathcal{F}_I \varphi)$  *<proof>*

Frontier operator as derived from closure.

**definition** *Fr-cl*:: $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{F}_C$ ) **where**  $\mathcal{F}_C \mathcal{C} \equiv \lambda A. (\mathcal{C} A) \wedge \mathcal{C}(-A)$

**lemma** *FC1*: *Cl-1*  $\varphi \implies \text{Cl-2 } \varphi \implies \text{Fr-1}(\mathcal{F}_C \varphi)$  *<proof>*

**lemma** *FC2*: *Fr-2*( $\mathcal{F}_C \varphi$ ) *<proof>*

**lemma** *FC3*: *Cl-3*  $\varphi \implies \text{Fr-3}(\mathcal{F}_C \varphi)$  *<proof>*

**lemma** *FC4*: *Cl-1b*  $\varphi \implies \text{Cl-2 } \varphi \implies \text{Cl-4 } \varphi \implies \text{Fr-4}(\mathcal{F}_C \varphi)$  *<proof>*

Frontier operator as derived from border.

**definition** *Fr-br*:: $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{F}_B$ ) **where**  $\mathcal{F}_B \mathcal{B} \equiv \lambda A. \mathcal{B} A \vee \mathcal{B}(-A)$

**lemma** *FB1*: *Br-1*  $\varphi \implies \text{Fr-1}(\mathcal{F}_B \varphi)$  *<proof>*

**lemma** *FB2*: *Fr-2*( $\mathcal{F}_B \varphi$ ) *<proof>*

**lemma** *FB3*: *Br-1*  $\varphi \implies \text{Br-2 } \varphi \implies \text{Fr-3}(\mathcal{F}_B \varphi)$  *<proof>*

**lemma FB4:**  $Br-1 \varphi \implies Br-3 \varphi \implies Fr-4(\mathcal{F}_B \varphi) \langle proof \rangle$

Border operator as derived from interior.

**definition Br-int::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{B}_I)$  **where**  $\mathcal{B}_I \mathcal{I} \equiv \lambda A. A \leftarrow (\mathcal{I} A)$

**lemma BI1:**  $Int-1 \varphi \implies Br-1(\mathcal{B}_I \varphi) \langle proof \rangle$

**lemma BI2:**  $Int-3 \varphi \implies Br-2(\mathcal{B}_I \varphi) \langle proof \rangle$

**lemma BI3:**  $Int-1a \varphi \implies Int-4 \varphi \implies Br-3(\mathcal{B}_I \varphi) \langle proof \rangle$

Border operator as derived from closure.

**definition Br-cl::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{B}_C)$  **where**  $\mathcal{B}_C \mathcal{C} \equiv \lambda A. A \wedge \mathcal{C}(-A)$

**lemma BC1:**  $Cl-1 \varphi \implies Br-1(\mathcal{B}_C \varphi) \langle proof \rangle$

**lemma BC2:**  $Cl-3 \varphi \implies Br-2(\mathcal{B}_C \varphi) \langle proof \rangle$

**lemma BC3:**  $Cl-1b \varphi \implies Cl-4 \varphi \implies Br-3(\mathcal{B}_C \varphi) \langle proof \rangle$

Note that the previous two conversion functions are related:

**lemma BI-BC-rel:**  $(\mathcal{B}_I \varphi) = \mathcal{B}_C(\varphi^d) \langle proof \rangle$

Border operator as derived from frontier.

**definition Br-fr::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{B}_F)$  **where**  $\mathcal{B}_F \mathcal{F} \equiv \lambda A. A \wedge (\mathcal{F} A)$

**lemma BF1:**  $Fr-1 \varphi \implies Br-1(\mathcal{B}_F \varphi) \langle proof \rangle$

**lemma BF2:**  $Fr-2 \varphi \implies Fr-3 \varphi \implies Br-2(\mathcal{B}_F \varphi) \langle proof \rangle$

**lemma BF3:**  $Fr-1b \varphi \implies Fr-2 \varphi \implies Fr-4 \varphi \implies Br-3(\mathcal{B}_F \varphi) \langle proof \rangle$

Interior operator as derived from border.

**definition Int-br::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{I}_B)$  **where**  $\mathcal{I}_B \mathcal{B} \equiv \lambda A. A \leftarrow (\mathcal{B} A)$

**lemma IB1:**  $Br-1 \varphi \implies Int-1(\mathcal{I}_B \varphi) \langle proof \rangle$

**lemma IB2:**  $Int-2(\mathcal{I}_B \varphi) \langle proof \rangle$

**lemma IB3:**  $Br-2 \varphi \implies Int-3(\mathcal{I}_B \varphi) \langle proof \rangle$

**lemma IB4:**  $Br-1 \varphi \implies Br-3 \varphi \implies Int-4(\mathcal{I}_B \varphi) \langle proof \rangle$

Interior operator as derived from frontier.

**definition Int-fr::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{I}_F)$  **where**  $\mathcal{I}_F \mathcal{F} \equiv \lambda A. A \leftarrow (\mathcal{F} A)$

**lemma IF1a:**  $Fr-1b \varphi \implies Int-1a(\mathcal{I}_F \varphi) \langle proof \rangle$

**lemma IF1b:**  $Fr-1a \varphi \implies Int-1b(\mathcal{I}_F \varphi) \langle proof \rangle$

**lemma IF1:**  $Fr-1 \varphi \implies Int-1(\mathcal{I}_F \varphi) \langle proof \rangle$

**lemma IF2:**  $Int-2(\mathcal{I}_F \varphi) \langle proof \rangle$

**lemma IF3:**  $Fr-2 \varphi \implies Fr-3 \varphi \implies Int-3(\mathcal{I}_F \varphi) \langle proof \rangle$

**lemma IF4:**  $Fr-1a \varphi \implies Fr-2 \varphi \implies Fr-4 \varphi \implies Int-4(\mathcal{I}_F \varphi) \langle proof \rangle$

Closure operator as derived from border.

**definition Cl-br::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{C}_B)$  **where**  $\mathcal{C}_B \mathcal{B} \equiv \lambda A. A \vee \mathcal{B}(-A)$

**lemma CB1:**  $Br-1 \varphi \implies Cl-1(\mathcal{C}_B \varphi) \langle proof \rangle$

**lemma CB2:**  $Cl-2(\mathcal{C}_B \varphi) \langle proof \rangle$

**lemma CB3:**  $Br-2 \varphi \implies Cl-3(\mathcal{C}_B \varphi) \langle proof \rangle$

**lemma CB4:**  $Br-1 \varphi \implies Br-3 \varphi \implies Cl-4(\mathcal{C}_B \varphi) \langle proof \rangle$

Closure operator as derived from frontier.

**definition Cl-fr::** $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma) (\mathcal{C}_F)$  **where**  $\mathcal{C}_F \mathcal{F} \equiv \lambda A. A \vee (\mathcal{F} A)$

**lemma CF1b:**  $Fr-1b \varphi \implies Fr-2 \varphi \implies Cl-1b(\mathcal{C}_F \varphi) \langle proof \rangle$

**lemma CF1a:**  $Fr-1a \varphi \implies Fr-2 \varphi \implies Cl-1a(\mathcal{C}_F \varphi) \langle proof \rangle$

**lemma CF1:**  $Fr-1 \varphi \implies Fr-2 \varphi \implies Cl-1(\mathcal{C}_F \varphi) \langle proof \rangle$

**lemma CF2:**  $Cl-2(\mathcal{C}_F \varphi) \langle proof \rangle$

**lemma CF3:**  $Fr-3 \varphi \implies Cl-3(\mathcal{C}_F \varphi) \langle proof \rangle$

**lemma CF4:**  $Fr-1a \varphi \implies Fr-2 \varphi \implies Fr-4 \varphi \implies Cl-4(\mathcal{C}_F \varphi) \langle proof \rangle$

#### 4.2.4 Infinitary conditions

We define the essential infinitary conditions for the closure and interior operators (entailing infinite additivity and multiplicativity resp.). Observe that the other direction is implied by monotonicity (MONO).

**abbreviation** *Cl-inf*  $\varphi \equiv iADDI-a(\varphi)$

**abbreviation** *Int-inf*  $\varphi \equiv iMULT-b(\varphi)$

There exists indeed a condition on frontier operators responsible for the infinitary conditions above:

**definition** *Fr-inf*  $\varphi \equiv \forall S. \bigwedge S \wedge \varphi(\bigwedge S) \preceq \bigwedge S \wedge \bigvee Ra[\varphi|S]$

**lemma** *CF-inf*: *Fr-2*  $\varphi \implies$  *Fr-inf*  $\varphi \implies$  *Cl-inf*  $(\mathcal{C}_F \varphi)$  *<proof>*

**lemma** *IF-inf*: *Fr-inf*  $\varphi \implies$  *Int-inf*  $(\mathcal{I}_F \varphi)$  *<proof>*

This condition is indeed strong enough to entail closure of the fixed-point predicates under infimum/supremum.

**lemma** *fp-IF-inf-closed*: *Fr-inf*  $\varphi \implies$  *infimum-closed*  $(fp(\mathcal{I}_F \varphi))$  *<proof>*

**lemma** *fp-CF-sup-closed*: *Fr-inf*  $\varphi \implies$  *Fr-2*  $\varphi \implies$  *supremum-closed*  $(fp(\mathcal{C}_F \varphi))$  *<proof>*

**end**

**theory** *topo-operators-derivative*

**imports** *topo-operators-basic*

**begin**

**nitpick-params** $[assms=true, user-axioms=true, show-all, expect=genuine, format=3]$

### 4.3 Derivative and coherence

In this section we investigate two related operators, namely the ‘derivative’ (or ‘derived set’) and the (Cantorian) ‘coherence’ of a set. The derivative of a set is the set of its accumulation (aka. limit) points. The coherence of a set A is the set formed by those limit points of A belonging to A. For the derivative operator we draw upon the works by Kuratowski [8] and (in more detail) by Zarycki [14]; cf. also McKinsey & Tarski [11]. For the (Cantorian) coherence operator we follow the treatment given by Zarycki in [13].

#### 4.3.1 Derivative conditions

The derivative conditions overlap partly with Kuratowski closure conditions [8]. We try to make both notations coincide when possible.

**abbreviation** *Der-1*  $\varphi \equiv Cl-1 \varphi$

**abbreviation** *Der-1a*  $\varphi \equiv Cl-1a \varphi$

**abbreviation** *Der-1b*  $\varphi \equiv Cl-1b \varphi$

**abbreviation** *Der-2*  $\varphi \equiv Cl-5 \varphi$  — follows from Cl-2

**abbreviation** *Der-3*  $\varphi \equiv Cl-3 \varphi$

**abbreviation** *Der-4*  $\varphi \equiv Cl-4' \varphi$

**definition** *Der-4e*  $\varphi \equiv \forall A. \varphi(\varphi A) \preceq (\varphi A \vee A)$

**definition** *Der-5*  $\varphi \equiv \forall A. (\varphi A \preceq A) \wedge (A \preceq \varphi^d A) \longrightarrow (A \approx \perp \vee A \approx \top)$

Some remarks: Condition Der-2 basically says (when assuming other derivative axioms) that the space is dense-in-itself, i.e. that all points are accumulation points (no point is isolated) w.r.t the whole space. Der-4 is a weakened (left-to-right) variant of Cl-4. Condition Der-4e corresponds to a (weaker) condition than Der-4 and is used in more recent literature (in particular in the

works of Leo Esakia [5]). When other derivative axioms are assumed, Der-5 above as used by Zarycki [14] says that the only clopen sets in the space are the top and bottom elements (empty set and universe, resp.). We verify some properties:

- lemma** *Der4e-rel*:  $Der-4 \varphi \implies Der-4e \varphi$  *<proof>*  
**lemma** *PD1*:  $Der-1b \varphi \implies \forall A B. A \preceq B \implies \varphi A \preceq \varphi B$  *<proof>*  
**lemma** *PD2*:  $Der-1b \varphi \implies \forall A B. A \preceq B \implies \varphi^d A \preceq \varphi^d B$  *<proof>*  
**lemma** *PD3*:  $Der-1b \varphi \implies \forall A B. \varphi(A \wedge B) \preceq \varphi A \wedge \varphi B$  *<proof>*  
**lemma** *PD4*:  $Der-1 \varphi \implies \forall A B. (\varphi A \leftarrow \varphi B) \preceq \varphi(A \leftarrow B)$  *<proof>*  
**lemma** *PD5*:  $Der-4 \varphi \implies \forall A. \varphi(\varphi(-(\varphi A))) \preceq \varphi(-(\varphi A))$  *<proof>*

Observe that the lemmas below require Der-2 as premise:

- lemma** *PD6*:  $Der-1 \varphi \implies Der-2 \varphi \implies \forall A. \varphi^d A \preceq \varphi A$  *<proof>*  
**lemma** *PD7*:  $Der-1 \varphi \implies Der-2 \varphi \implies \forall A. \varphi(\varphi^d A) \preceq \varphi(\varphi A)$  *<proof>*  
**lemma** *PD8*:  $Der-1 \varphi \implies Der-2 \varphi \implies Der-4 \varphi \implies \forall A. \varphi(\varphi^d A) \preceq \varphi A$  *<proof>*  
**lemma** *PD9*:  $Der-1 \varphi \implies Der-2 \varphi \implies Der-4 \varphi \implies \forall A. \varphi^d A \preceq \varphi^d(\varphi A)$  *<proof>*  
**lemma** *PD10*:  $Der-1 \varphi \implies Der-2 \varphi \implies Der-4 \varphi \implies \forall A. \varphi^d A \preceq \varphi(\varphi^d A)$  *<proof>*  
**lemma** *PD11*:  $Der-1 \varphi \implies Der-2 \varphi \implies Der-4 \varphi \implies \forall A. -(\varphi A) \preceq \varphi(-(\varphi A))$  *<proof>*  
**lemma** *PD12*:  $Der-1 \varphi \implies Der-2 \varphi \implies Der-4 \varphi \implies \forall A. (\varphi^d A) \wedge (\varphi A) \approx \varphi^d(A \wedge (\varphi A))$  *<proof>*

The conditions below can serve to axiomatize a derivative operator. Different authors consider different sets of conditions. We define below some corresponding to Zarycki [14], Kuratowski [8] [13], McKinsey & Tarski [11], and Esakia [5], respectively.

- abbreviation**  $\mathfrak{D}z \varphi \equiv Der-1 \varphi \wedge Der-2 \varphi \wedge Der-3 \varphi \wedge Der-4 \varphi \wedge Der-5 \varphi$   
**abbreviation**  $\mathfrak{D}k \varphi \equiv Der-1 \varphi \wedge Der-2 \varphi \wedge Der-3 \varphi \wedge Der-4 \varphi$   
**abbreviation**  $\mathfrak{D}mt \varphi \equiv Der-1 \varphi \wedge Der-3 \varphi \wedge Der-4 \varphi$   
**abbreviation**  $\mathfrak{D}e \varphi \equiv Der-1 \varphi \wedge Der-3 \varphi \wedge Der-4e \varphi$

Our ‘default’ derivative operator will coincide with  $\mathfrak{D}k$  from Kuratowski (also Zarycki). However, for proving theorems we will employ the weaker variant Der-4e instead of Der-4 whenever possible. We start by defining a dual operator and verifying some dualities; we then define conversion operators. Observe that conditions Der-2 and Der-5 are not used in the rest of this subsection. Der-2 will be required later when working with the coherence operator.

- abbreviation**  $\mathfrak{D} \varphi \equiv \mathfrak{D}k \varphi$   
**abbreviation**  $\Sigma \varphi \equiv Int-1 \varphi \wedge Int-3 \varphi \wedge Int-4' \varphi \wedge Int-5 \varphi$  — ‘dual-derivative’ operator

- lemma** *SD-dual1*:  $\Sigma(\varphi) \implies \mathfrak{D}(\varphi^d)$  *<proof>*  
**lemma** *SD-dual2*:  $\Sigma(\varphi^d) \implies \mathfrak{D}(\varphi)$  *<proof>*  
**lemma** *DS-dual1*:  $\mathfrak{D}(\varphi) \implies \Sigma(\varphi^d)$  *<proof>*  
**lemma** *DS-dual2*:  $\mathfrak{D}(\varphi^d) \implies \Sigma(\varphi)$  *<proof>*  
**lemma** *DS-dual*:  $\mathfrak{D}(\varphi) = \Sigma(\varphi^d)$  *<proof>*

Closure operator as derived from derivative.

- definition** *Cl-der*:: $(\sigma \implies \sigma) \implies (\sigma \implies \sigma)$  ( $\mathcal{C}_D \mathcal{D}$ ) **where**  $\mathcal{C}_D \mathcal{D} \equiv \lambda A. A \vee \mathcal{D}(A)$

Verify properties:

- lemma** *CD1a*:  $Der-1a \mathcal{D} \implies Cl-1a (\mathcal{C}_D \mathcal{D})$  *<proof>*  
**lemma** *CD1b*:  $Der-1b \mathcal{D} \implies Cl-1b (\mathcal{C}_D \mathcal{D})$  *<proof>*  
**lemma** *CD1* :  $Der-1 \mathcal{D} \implies Cl-1 (\mathcal{C}_D \mathcal{D})$  *<proof>*  
**lemma** *CD2*:  $Cl-2 (\mathcal{C}_D \mathcal{D})$  *<proof>*  
**lemma** *CD3*:  $Der-3 \mathcal{D} \implies Der-3 (\mathcal{C}_D \mathcal{D})$  *<proof>*  
**lemma** *CD4a*:  $Der-1a \mathcal{D} \implies Der-4e \mathcal{D} \implies Cl-4 (\mathcal{C}_D \mathcal{D})$  *<proof>*  
**lemma** *Der-1b*  $\mathcal{D} \implies Der-4 \mathcal{D} \implies Cl-4 (\mathcal{C}_D \mathcal{D})$  **nitpick** *<proof>*  
**lemma** *CD4*:  $Der-1 \mathcal{D} \implies Der-4e \mathcal{D} \implies Cl-4 (\mathcal{C}_D \mathcal{D})$  *<proof>*



Interior operator as derived from (dual) derivative.

**definition**  $Int\text{-}der::(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{I}_D$ ) **where**  $\mathcal{I}_D \mathcal{D} \equiv \lambda A. A \wedge \mathcal{D}^d(A)$

Verify definition:

**lemma**  $Int\text{-}der\text{-}def2: \mathcal{I}_D \mathcal{D} = (\lambda \varphi. \varphi \leftarrow \mathcal{D}(-\varphi))$   $\langle proof \rangle$

**lemma**  $dual\text{-}der1: \mathcal{C}_D \mathcal{D} \equiv (\mathcal{I}_D \mathcal{D})^d$   $\langle proof \rangle$

**lemma**  $dual\text{-}der2: \mathcal{I}_D \mathcal{D} \equiv (\mathcal{C}_D \mathcal{D})^d$   $\langle proof \rangle$

Verify properties:

**lemma**  $ID1: Der\text{-}1 \mathcal{D} \Longrightarrow Int\text{-}1 (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $ID1a: Der\text{-}1a \mathcal{D} \Longrightarrow Int\text{-}1b (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $ID1b: Der\text{-}1b \mathcal{D} \Longrightarrow Int\text{-}1a (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $ID2: Int\text{-}2 (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $ID3: Der\text{-}3 \mathcal{D} \Longrightarrow Int\text{-}3 (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $ID4: Der\text{-}1 \mathcal{D} \Longrightarrow Der\text{-}4e \mathcal{D} \Longrightarrow Int\text{-}4 (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $ID4a: Der\text{-}1a \mathcal{D} \Longrightarrow Der\text{-}4e \mathcal{D} \Longrightarrow Int\text{-}4 (\mathcal{I}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $Der\text{-}1b \mathcal{D} \Longrightarrow Der\text{-}4 \mathcal{D} \Longrightarrow Int\text{-}4 (\mathcal{I}_D \mathcal{D})$  **nitpick**  $\langle proof \rangle$

Border operator as derived from (dual) derivative.

**definition**  $Br\text{-}der::(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{B}_D$ ) **where**  $\mathcal{B}_D \mathcal{D} \equiv \lambda A. A \leftarrow \mathcal{D}^d(A)$

Verify definition:

**lemma**  $Br\text{-}der\text{-}def2: \mathcal{B}_D \mathcal{D} = (\lambda A. A \wedge \mathcal{D}(-A))$   $\langle proof \rangle$

Verify properties:

**lemma**  $BD1: Der\text{-}1 \mathcal{D} \Longrightarrow Br\text{-}1 (\mathcal{B}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $BD2: Der\text{-}3 \mathcal{D} \Longrightarrow Br\text{-}2 (\mathcal{B}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $BD3: Der\text{-}1b \mathcal{D} \Longrightarrow Der\text{-}4e \mathcal{D} \Longrightarrow Br\text{-}3 (\mathcal{B}_D \mathcal{D})$   $\langle proof \rangle$

Frontier operator as derived from derivative.

**definition**  $Fr\text{-}der::(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{F}_D$ ) **where**  $\mathcal{F}_D \mathcal{D} \equiv \lambda A. (A \leftarrow \mathcal{D}^d(A)) \vee (\mathcal{D}(A) \leftarrow A)$

Verify definition:

**lemma**  $Fr\text{-}der\text{-}def2: \mathcal{F}_D \mathcal{D} = (\lambda A. (A \vee \mathcal{D}(A)) \wedge (-A \vee \mathcal{D}(-A)))$   $\langle proof \rangle$

Verify properties:

**lemma**  $FD1a: Der\text{-}1a \mathcal{D} \Longrightarrow Fr\text{-}1a(\mathcal{F}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $FD1b: Der\text{-}1b \mathcal{D} \Longrightarrow Fr\text{-}1b(\mathcal{F}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $FD1: Der\text{-}1 \mathcal{D} \Longrightarrow Fr\text{-}1(\mathcal{F}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $FD2: Fr\text{-}2(\mathcal{F}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $FD3: Der\text{-}3 \mathcal{D} \Longrightarrow Fr\text{-}3(\mathcal{F}_D \mathcal{D})$   $\langle proof \rangle$

**lemma**  $FD4: Der\text{-}1 \mathcal{D} \Longrightarrow Der\text{-}4e \mathcal{D} \Longrightarrow Fr\text{-}4(\mathcal{F}_D \mathcal{D})$   $\langle proof \rangle$

Note that the derivative operation cannot be obtained from interior, closure, border, nor frontier. In this respect the derivative is a fundamental operation.

### 4.3.2 Infinitary conditions

The corresponding infinitary condition on derivative operators is inherited from the one for closure.

**abbreviation**  $Der\text{-}inf \varphi \equiv Cl\text{-}inf(\varphi)$

**lemma** *CD-inf*:  $Der\text{-}inf\ \varphi \implies Cl\text{-}inf(\mathcal{C}_D\ \varphi)$  *<proof>*

**lemma** *ID-inf*:  $Der\text{-}inf\ \varphi \implies Int\text{-}inf(\mathcal{I}_D\ \varphi)$  *<proof>*

This condition is indeed strong enough as to entail closure of some fixed-point predicates under infimum/supremum.

**lemma** *fp-ID-inf-closed*:  $Der\text{-}inf\ \varphi \implies infimum\text{-}closed\ (fp\ (\mathcal{I}_D\ \varphi))$  *<proof>*

**lemma** *fp-CD-sup-closed*:  $Der\text{-}inf\ \varphi \implies supremum\text{-}closed\ (fp\ (\mathcal{C}_D\ \varphi))$  *<proof>*

### 4.3.3 Coherence conditions

We finish this section by introducing the ‘coherence’ operator (Cantor’s ‘Koherenz’) as discussed by Zarycki in [13]. As happens with the derivative operator, the coherence operator cannot be derived from interior, closure, border, nor frontier.

**definition** *Kh-1*  $\varphi \equiv ADDI\text{-}b\ \varphi$

**definition** *Kh-2*  $\varphi \equiv dEXP\ \varphi$

**definition** *Kh-3*  $\varphi \equiv \forall A. \varphi(\varphi^d A) \approx \varphi^d(\varphi A)$

**lemma** *PK1*: *Kh-1*  $\varphi \equiv MONO\ \varphi$  *<proof>*

**lemma** *PK2*: *Kh-1*  $\varphi \equiv \forall A B. \varphi(A \wedge B) \preceq (\varphi A) \wedge (\varphi B)$  *<proof>*

**lemma** *PK3*: *Kh-2*  $\varphi \implies \varphi \perp \approx \perp$  *<proof>*

**lemma** *PK4*: *Kh-1*  $\varphi \implies Kh\text{-}3\ \varphi \implies \varphi \top \approx \top$  *<proof>*

**lemma** *PK5*: *Kh-2*  $\varphi \implies \forall A. \varphi(\neg A) \preceq \neg(\varphi A)$  *<proof>*

**lemma** *PK6*: *Kh-1*  $\varphi \implies Kh\text{-}2\ \varphi \implies \forall A B. \varphi(A \leftarrow B) \preceq (\varphi A) \leftarrow (\varphi B)$  *<proof>*

**lemma** *PK7*: *Kh-3*  $\varphi \implies \forall A. \varphi(\varphi(\neg(\varphi A))) \approx \varphi(\neg(\varphi(\varphi^d A)))$  *<proof>*

**lemma** *PK8*: *Kh-3*  $\varphi \implies \forall A. \varphi(\neg(\varphi(\varphi A))) \approx \varphi^d(\varphi(\neg(\varphi A)))$  *<proof>*

Coherence operator as derived from derivative (requires conditions Der-2 and Der-4).

**definition** *Kh-der*:  $(\sigma \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{K}_D$ ) **where**  $\mathcal{K}_D\ \mathcal{D} \equiv \lambda A. A \wedge (\mathcal{D}\ A)$

Verify properties:

**lemma** *KD1*: *Der-1*  $\varphi \implies Kh\text{-}1\ (\mathcal{K}_D\ \varphi)$  *<proof>*

**lemma** *KD2*: *Kh-2* ( $\mathcal{K}_D\ \varphi$ ) *<proof>*

**lemma** *KD3*: *Der-1*  $\varphi \implies Der\text{-}2\ \varphi \implies Der\text{-}4\ \varphi \implies Kh\text{-}3\ (\mathcal{K}_D\ \varphi)$  *<proof>*

**end**

**theory** *topo-alexandrov*

**imports** *sse-operation-positive-quantification*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 5 Generalized specialization orderings and Alexandrov topologies

A topology is called ‘Alexandrov’ (after the Russian mathematician Pavel Alexandrov) if the intersection (resp. union) of any (finite or infinite) family of open (resp. closed) sets is open (resp. closed); in algebraic terms, this means that the set of fixed points of the interior (closure) operation is closed under infinite meets (joins). Another common algebraic formulation requires the closure (interior) operation to satisfy the infinitary variants of additivity (multiplicativity), i.e. iADDI (iMULT) as introduced before.

In the literature, the well-known Kuratowski conditions for the closure (resp. interior) operation are assumed, namely: ADDI, EXP, NOR, IDEM (resp. MULT, dEXP, dNOR, IDEM). This

makes both formulations equivalent. However, this is not the case in general if those conditions become negotiable.

Alexandrov topologies have interesting properties relating them to the semantics of modal logic. Assuming Kuratowski conditions, Alexandrov topological operations defined on subsets of  $S$  are in one-to-one correspondence with preorders on  $S$ ; in topological terms, Alexandrov topologies are uniquely determined by their specialization preorders. Since we do not presuppose any Kuratowski conditions to begin with, the preorders in question are in general not even transitive. Here we just call them 'specialization relations'. We will still call (generalized) closure/interior-like operations as such (for lack of a better name). We explore minimal conditions under which some relevant results for the semantics of modal logic obtain.

## 5.1 Specialization relations

Specialization relations (among worlds/points) are particular cases of propositional functions with type  $w \Rightarrow \sigma$ .

Define some relevant properties of relations:

**abbreviation** *serial*  $R \equiv \forall x. \exists y. R x y$

**abbreviation** *reflexive*  $R \equiv \forall x. R x x$

**abbreviation** *transitive*  $R \equiv \forall x y z. R x y \wedge R y z \longrightarrow R x z$

**abbreviation** *antisymmetric*  $R \equiv \forall x y. R x y \wedge R y x \longrightarrow x = y$

**abbreviation** *symmetric*  $R \equiv \forall x y. R x y \longrightarrow R y x$

Closure/interior operations can be derived from an arbitrary relation as operations returning down-/up-sets.

**definition** *Cl-rel*:: $(w \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{C}_R$ ) **where**  $\mathcal{C}_R R \equiv \lambda A. \lambda w. \exists v. R w v \wedge A v$

**definition** *Int-rel*:: $(w \Rightarrow \sigma) \Rightarrow (\sigma \Rightarrow \sigma)$  ( $\mathcal{I}_R$ ) **where**  $\mathcal{I}_R R \equiv \lambda A. \lambda w. \forall v. R w v \longrightarrow A v$

Duality between interior and closure follows directly:

**lemma** *dual-rel1*:  $\forall A. (\mathcal{C}_R R) A \approx (\mathcal{I}_R R)^d A$  *<proof>*

**lemma** *dual-rel2*:  $\forall A. (\mathcal{I}_R R) A \approx (\mathcal{C}_R R)^d A$  *<proof>*

We explore minimal conditions of the specialization relation under which some operation's conditions obtain.

**lemma** *rC1*: *ADDI* ( $\mathcal{C}_R R$ ) *<proof>*

**lemma** *rC1i*: *iADDI* ( $\mathcal{C}_R R$ ) *<proof>*

**lemma** *rC2*: *reflexive*  $R \longrightarrow \text{EXP}$  ( $\mathcal{C}_R R$ ) *<proof>*

**lemma** *rC3*: *NOR* ( $\mathcal{C}_R R$ ) *<proof>*

**lemma** *rC4*: *reflexive*  $R \wedge$  *transitive*  $R \longrightarrow \text{IDEM}$  ( $\mathcal{C}_R R$ ) *<proof>*

**lemma** *rC-Barcan*:  $\forall \pi. (\mathcal{C}_R R)(\exists x. \pi x) \preceq (\exists x. (\mathcal{C}_R R)(\pi x))$  *<proof>*

**lemma** *rI1*: *MULT* ( $\mathcal{I}_R R$ ) *<proof>*

**lemma** *rI1i*: *iMULT* ( $\mathcal{I}_R R$ ) *<proof>*

**lemma** *rI2*: *reflexive*  $R \implies \text{dEXP}$  ( $\mathcal{I}_R R$ ) *<proof>*

**lemma** *rI3*: *dNOR* ( $\mathcal{I}_R R$ ) *<proof>*

**lemma** *rI4*: *reflexive*  $R \implies$  *transitive*  $R \implies \text{IDEM}$  ( $\mathcal{I}_R R$ ) *<proof>*

**lemma** *rI-Barcan*:  $\forall \pi. (\forall x. (\mathcal{I}_R R)(\pi x)) \preceq (\mathcal{I}_R R)(\forall x. \pi x)$  *<proof>*

A specialization relation can be derived from a given operation (intended as a closure-like operation).

**definition** *sp-rel*:: $(\sigma \Rightarrow \sigma) \Rightarrow (w \Rightarrow \sigma)$  ( $\mathcal{R}^C$ ) **where**  $\mathcal{R}^C C \equiv \lambda w v. C (\lambda u. u=v) w$

Preorder properties of the specialization relation follow directly from the corresponding operation's conditions.

**lemma** *sp-rel-reflex*:  $EXP \mathcal{C} \implies \text{reflexive } (\mathcal{R}^C \mathcal{C}) \langle \text{proof} \rangle$

**lemma** *sp-rel-trans*:  $MONO \mathcal{C} \implies IDEM \mathcal{C} \implies \text{transitive } (\mathcal{R}^C \mathcal{C}) \langle \text{proof} \rangle$

However, we can obtain finite countermodels for antisymmetry and symmetry given all relevant conditions. We will revisit this issue later and examine their relation with the topological separation axioms T0 and T1 resp.

**lemma** *iADDI*  $\mathcal{C} \implies EXP \mathcal{C} \implies NOR \mathcal{C} \implies IDEM \mathcal{C} \implies \text{antisymmetric } (\mathcal{R}^C \mathcal{C})$  **nitpick**  $\langle \text{proof} \rangle$

**lemma** *iADDI*  $\mathcal{C} \implies EXP \mathcal{C} \implies NOR \mathcal{C} \implies IDEM \mathcal{C} \implies \text{symmetric } (\mathcal{R}^C \mathcal{C})$  **nitpick**  $\langle \text{proof} \rangle$

## 5.2 Alexandrov topology

As mentioned previously, Alexandrov closure (and by duality interior) operations correspond to specialization relations. It is worth mentioning that in Alexandrov topologies every point has a minimal/smallest neighborhood, namely the set of points related to it by the specialization (aka. accessibility) relation. Alexandrov spaces are thus also called 'finitely generated'. We examine below minimal conditions under which these relations obtain.

**lemma** *sp-rel-a*:  $MONO \mathcal{C} \implies \forall A. (\mathcal{C}_R (\mathcal{R}^C \mathcal{C})) A \preceq \mathcal{C} A \langle \text{proof} \rangle$

**lemma** *sp-rel-b*: *iADDI-a*  $\mathcal{C} \implies \forall A. (\mathcal{C}_R (\mathcal{R}^C \mathcal{C})) A \succeq \mathcal{C} A \langle \text{proof} \rangle$

**lemma** *sp-rel*: *iADDI*  $\mathcal{C} \implies \forall A. \mathcal{C} A \approx (\mathcal{C}_R (\mathcal{R}^C \mathcal{C})) A \langle \text{proof} \rangle$

It is instructive to expand the definitions in the above lemma:

**lemma** *iADDI*  $\mathcal{C} \implies \forall A. \forall w. (\mathcal{C} A) w \longleftrightarrow (\exists v. A v \wedge (\mathcal{C} (\lambda u. u=v)) w) \langle \text{proof} \rangle$

We now turn to the more traditional characterization of Alexandrov topologies in terms of closure under infinite joins/meets.

Fixed points of operations satisfying ADDI (MULT) are not in general closed under infinite joins (meets). For the given conditions countermodels are expected to be infinite. We (sanity) check that nitpick cannot find any.

**lemma** *ADDI*( $\varphi$ )  $\implies \text{supremum-closed } (fp \varphi) \langle \text{proof} \rangle$

**lemma** *MULT*( $\varphi$ )  $\implies \text{infimum-closed } (fp \varphi) \langle \text{proof} \rangle$

By contrast, we can show that this obtains if assuming the corresponding infinitary variants (iADDI/iMULT).

**lemma** *iADDI*( $\varphi$ )  $\implies \text{supremum-closed } (fp \varphi) \langle \text{proof} \rangle$

**lemma** *iMULT*( $\varphi$ )  $\implies \text{infimum-closed } (fp \varphi) \langle \text{proof} \rangle$

As shown above, closure (interior) operations derived from relations readily satisfy iADDI (iMULT), being thus closed under infinite joins (meets).

**lemma** *supremum-closed* ( $fp (\mathcal{C}_R R)$ )  $\langle \text{proof} \rangle$

**lemma** *infimum-closed* ( $fp (\mathcal{I}_R R)$ )  $\langle \text{proof} \rangle$

## 5.3 (Anti)symmetry and the separation axioms T0 and T1

We can now revisit the relationship between (anti)symmetry and the separation axioms T1 and T0.

T0: any two distinct points in the space can be separated by an open set (i.e. containing one point and not the other).

**abbreviation**  $T0\text{-sep } \mathcal{C} \equiv \forall w v. w \neq v \longrightarrow (\exists G. (fp \mathcal{C}^d)(G) \wedge (G w \neq G v))$

T1: any two distinct points can be separated by (two not necessarily disjoint) open sets, i.e. all singletons are closed.

**abbreviation**  $T1\text{-sep } \mathcal{C} \equiv \forall w. (fp \mathcal{C})(\lambda u. u = w)$

We can (sanity) check that T1 entails T0 but not viceversa.

**lemma**  $T0\text{-sep } \mathcal{C} \implies T1\text{-sep } \mathcal{C}$  **nitpick**  $\langle proof \rangle$

**lemma**  $T1\text{-sep } \mathcal{C} \implies T0\text{-sep } \mathcal{C}$   $\langle proof \rangle$

Under appropriate conditions, T0-separation corresponds to antisymmetry of the specialization relation (here an ordering).

**lemma**  $T0\text{-sep } \mathcal{C} \iff antisymmetric (\mathcal{R}^C \mathcal{C})$  **nitpick**  $\langle proof \rangle$

**lemma**  $T0\text{-antisymm-a: } MONO \mathcal{C} \implies T0\text{-sep } \mathcal{C} \longrightarrow antisymmetric (\mathcal{R}^C \mathcal{C})$   $\langle proof \rangle$

**lemma**  $T0\text{-antisymm-b: } EXP \mathcal{C} \implies IDEM \mathcal{C} \implies antisymmetric (\mathcal{R}^C \mathcal{C}) \longrightarrow T0\text{-sep } \mathcal{C}$   $\langle proof \rangle$

**lemma**  $T0\text{-antisymm: } MONO \mathcal{C} \implies EXP \mathcal{C} \implies IDEM \mathcal{C} \implies T0\text{-sep } \mathcal{C} = antisymmetric (\mathcal{R}^C \mathcal{C})$   $\langle proof \rangle$

Also, under the appropriate conditions, T1-separation corresponds to symmetry of the specialization relation.

**lemma**  $T1\text{-symm-a: } T1\text{-sep } \mathcal{C} \longrightarrow symmetric (\mathcal{R}^C \mathcal{C})$   $\langle proof \rangle$

**lemma**  $T1\text{-symm-b: } MONO \mathcal{C} \implies EXP \mathcal{C} \implies T0\text{-sep } \mathcal{C} \implies symmetric (\mathcal{R}^C \mathcal{C}) \longrightarrow T1\text{-sep } \mathcal{C}$   $\langle proof \rangle$

**lemma**  $T1\text{-symm: } MONO \mathcal{C} \implies EXP \mathcal{C} \implies T0\text{-sep } \mathcal{C} \implies symmetric (\mathcal{R}^C \mathcal{C}) = T1\text{-sep } \mathcal{C}$   $\langle proof \rangle$

**end**

**theory** *topo-frontier-algebra*

**imports** *topo-operators-basic*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 6 Frontier Algebra

The closure of a set A ( $\mathcal{C}(A)$ ) can be seen as the set A augmented by (i) its boundary points, or (ii) its accumulation/limit points. In this section we explore the first variant by drawing on the notion of a frontier algebra, defined in an analogous fashion as the well-known closure and interior algebras.

Declares a primitive (unconstrained) frontier (aka. boundary) operation and defines others from it.

**consts**  $\mathcal{F}::\sigma \Rightarrow \sigma$

**abbreviation**  $\mathcal{I} \equiv \mathcal{I}_F \mathcal{F}$  — interior

**abbreviation**  $\mathcal{C} \equiv \mathcal{C}_F \mathcal{F}$  — closure

**abbreviation**  $\mathcal{B} \equiv \mathcal{B}_F \mathcal{F}$  — border

### 6.1 Basic properties

Verifies minimal conditions under which operators resulting from conversion functions coincide.

**lemma**  $ICdual: Fr\text{-}2 \mathcal{F} \implies \mathcal{I} \equiv \mathcal{C}^d$   $\langle proof \rangle$

**lemma**  $ICdual': Fr\text{-}2 \mathcal{F} \implies \mathcal{C} \equiv \mathcal{I}^d$   $\langle proof \rangle$

**lemma**  $BI\text{-}rel: \mathcal{B} \equiv \mathcal{B}_I \mathcal{I}$   $\langle proof \rangle$

**lemma**  $IB\text{-}rel: \mathcal{I} \equiv \mathcal{I}_B \mathcal{B}$   $\langle proof \rangle$

**lemma**  $BC\text{-}rel: Fr\text{-}2 \mathcal{F} \implies \mathcal{B} \equiv \mathcal{B}_C \mathcal{C}$   $\langle proof \rangle$

**lemma** *CB-rel*:  $Fr-2 \mathcal{F} \implies \mathcal{C} \equiv \mathcal{C}_B \mathcal{B} \langle proof \rangle$   
**lemma** *FI-rel*:  $Fr-2 \mathcal{F} \implies \mathcal{F} \equiv \mathcal{F}_I \mathcal{I} \langle proof \rangle$   
**lemma** *FC-rel*:  $Fr-2 \mathcal{F} \implies \mathcal{F} \equiv \mathcal{F}_C \mathcal{C} \langle proof \rangle$   
**lemma** *FB-rel*:  $Fr-2 \mathcal{F} \implies \mathcal{F} \equiv \mathcal{F}_B \mathcal{B} \langle proof \rangle$

Fixed-point and other operators are interestingly related.

**lemma** *fp1*:  $\mathcal{I}^{fp} \equiv \mathcal{B}^c \langle proof \rangle$   
**lemma** *fp2*:  $\mathcal{B}^{fp} \equiv \mathcal{I}^c \langle proof \rangle$   
**lemma** *fp3*:  $Fr-2 \mathcal{F} \implies \mathcal{C}^{fp} \equiv \mathcal{B}^d \langle proof \rangle$   
**lemma** *fp4*:  $Fr-2 \mathcal{F} \implies (\mathcal{B}^d)^{fp} \equiv \mathcal{C} \langle proof \rangle$   
**lemma** *fp5*:  $\mathcal{F}^{fp} \equiv \mathcal{B} \sqcup (\mathcal{C}^c) \langle proof \rangle$

Different inter-relations (some redundant ones are kept to help the provers).

**lemma** *monI*:  $Fr-1b \mathcal{F} \implies MONO(\mathcal{I}) \langle proof \rangle$   
**lemma** *monC*:  $Fr-6 \mathcal{F} \implies MONO(\mathcal{C}) \langle proof \rangle$   
**lemma** *pB1*:  $\forall A. \mathcal{B} A \approx A \leftarrow \mathcal{I} A \langle proof \rangle$   
**lemma** *pB2*:  $\forall A. \mathcal{B} A \approx A \wedge \mathcal{F} A \langle proof \rangle$   
**lemma** *pB3*:  $Fr-2 \mathcal{F} \implies \forall A. \mathcal{B}(-A) \approx -A \wedge \mathcal{F} A \langle proof \rangle$   
**lemma** *pB4*:  $Fr-2 \mathcal{F} \implies \forall A. \mathcal{B}(-A) \approx -A \wedge \mathcal{C} A \langle proof \rangle$   
**lemma** *pB5*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies \forall A. \mathcal{B}(\mathcal{C} A) \preceq (\mathcal{B} A) \vee \mathcal{B}(-A) \langle proof \rangle$   
**lemma** *pF1*:  $\forall A. \mathcal{F} A \approx \mathcal{C} A \leftarrow \mathcal{I} A \langle proof \rangle$   
**lemma** *pF2*:  $Fr-2 \mathcal{F} \implies \forall A. \mathcal{F} A \approx \mathcal{C} A \wedge \mathcal{C}(-A) \langle proof \rangle$   
**lemma** *pF3*:  $Fr-2 \mathcal{F} \implies \forall A. \mathcal{F} A \approx \mathcal{B} A \vee \mathcal{B}(-A) \langle proof \rangle$   
**lemma** *pF4*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4(\mathcal{F}) \implies \forall A. \mathcal{F}(\mathcal{I} A) \preceq \mathcal{F} A \langle proof \rangle$   
**lemma** *pF5*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A. \mathcal{F}(\mathcal{C} A) \preceq \mathcal{F} A \langle proof \rangle$   
**lemma** *pA1*:  $\forall A. A \approx \mathcal{I} A \vee \mathcal{B} A \langle proof \rangle$   
**lemma** *pA2*:  $Fr-2 \mathcal{F} \implies \forall A. A \approx \mathcal{C} A \leftarrow \mathcal{B}(-A) \langle proof \rangle$   
**lemma** *pC1*:  $Fr-2 \mathcal{F} \implies \forall A. \mathcal{C} A \approx A \vee \mathcal{B}(-A) \langle proof \rangle$   
**lemma** *pC2*:  $\forall A. \mathcal{C} A \approx A \vee \mathcal{F} A \langle proof \rangle$   
**lemma** *pI1*:  $\forall A. \mathcal{I} A \approx A \leftarrow \mathcal{B} A \langle proof \rangle$   
**lemma** *pI2*:  $\forall A. \mathcal{I} A \approx A \leftarrow \mathcal{F} A \langle proof \rangle$

**lemma** *IC-imp*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall A B. \mathcal{I}(A \rightarrow B) \preceq \mathcal{C} A \rightarrow \mathcal{C} B \langle proof \rangle$

Defines some fixed-point predicates and prove some properties.

**abbreviation** *openset* (*Op*) **where**  $Op A \equiv fp \mathcal{I} A$   
**abbreviation** *closedset* (*Cl*) **where**  $Cl A \equiv fp \mathcal{C} A$   
**abbreviation** *borderset* (*Br*) **where**  $Br A \equiv fp \mathcal{B} A$   
**abbreviation** *frontierset* (*Fr*) **where**  $Fr A \equiv fp \mathcal{F} A$

**lemma** *Int-Open*:  $Fr-1a \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A. Op(\mathcal{I} A) \langle proof \rangle$   
**lemma** *Cl-Closed*:  $Fr-1a \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A. Cl(\mathcal{C} A) \langle proof \rangle$   
**lemma** *Br-Border*:  $Fr-1b \mathcal{F} \implies \forall A. Br(\mathcal{B} A) \langle proof \rangle$

In contrast, there is no analogous fixed-point result for frontier:

**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall A. Fr(\mathcal{F} A) \text{ nitpick} \langle proof \rangle$

**lemma** *OpClDual*:  $Fr-2 \mathcal{F} \implies \forall A. Cl A \longleftrightarrow Op(-A) \langle proof \rangle$   
**lemma** *ClOpDual*:  $Fr-2 \mathcal{F} \implies \forall A. Op A \longleftrightarrow Cl(-A) \langle proof \rangle$   
**lemma** *Fr-ClBr*:  $\forall A. Fr(A) = (Cl(A) \wedge Br(A)) \langle proof \rangle$   
**lemma** *Cl-F*:  $Fr-4 \mathcal{F} \implies \forall A. Cl(\mathcal{F} A) \langle proof \rangle$

## 6.2 Further properties

The definitions and theorems below are well known in the literature (e.g. [9]). Here we uncover the minimal conditions under which they hold (taking frontier operation as primitive).

**lemma** *Cl-Bzero*:  $Fr-2 \mathcal{F} \implies \forall A. Cl A \longleftrightarrow \mathcal{B}(-A) \approx \perp$  *<proof>*

**lemma** *Op-Bzero*:  $\forall A. Op A \longleftrightarrow (\mathcal{B} A) \approx \perp$  *<proof>*

**lemma** *Br-boundary*:  $Fr-2 \mathcal{F} \implies \forall A. Br(A) \longleftrightarrow \mathcal{I} A \approx \perp$  *<proof>*

**lemma** *Fr-nowhereDense*:  $Fr-2 \mathcal{F} \implies \forall A. Fr(A) \longrightarrow \mathcal{I}(C A) \approx \perp$  *<proof>*

**lemma** *Cl-FB*:  $\forall A. Cl A \longleftrightarrow \mathcal{F} A \approx \mathcal{B} A$  *<proof>*

**lemma** *Op-FB*:  $Fr-2 \mathcal{F} \implies \forall A. Op A \longleftrightarrow \mathcal{F} A \approx \mathcal{B}(-A)$  *<proof>*

**lemma** *Clopen-Fzero*:  $\forall A. Cl A \wedge Op A \longleftrightarrow \mathcal{F} A \approx \perp$  *<proof>*

**lemma** *Int-sup-closed*:  $Fr-1b \mathcal{F} \implies supremum-closed (\lambda A. Op A)$  *<proof>*

**lemma** *Int-meet-closed*:  $Fr-1a \mathcal{F} \implies meet-closed (\lambda A. Op A)$  *<proof>*

**lemma** *Int-inf-closed*:  $Fr-inf \mathcal{F} \implies infimum-closed (\lambda A. Op A)$  *<proof>*

**lemma** *Cl-inf-closed*:  $Fr-6 \mathcal{F} \implies infimum-closed (\lambda A. Cl A)$  *<proof>*

**lemma** *Cl-join-closed*:  $Fr-1a \mathcal{F} \implies Fr-2 \mathcal{F} \implies join-closed (\lambda A. Cl A)$  *<proof>*

**lemma** *Cl-sup-closed*:  $Fr-2 \mathcal{F} \implies Fr-inf \mathcal{F} \implies supremum-closed (\lambda A. Cl A)$  *<proof>*

**lemma** *Br-inf-closed*:  $Fr-1b \mathcal{F} \implies infimum-closed (\lambda A. Br A)$  *<proof>*

**lemma** *Fr-inf-closed*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies infimum-closed (\lambda A. Fr A)$  *<proof>*

**lemma** *Br-Fr-join*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A B. Br A \wedge Fr B \longrightarrow Br(A \vee B)$  *<proof>*

**lemma** *Fr-join-closed*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies join-closed (\lambda A. Fr A)$  *<proof>*

Introduces a predicate for indicating that two sets are disjoint and proves some properties.

**abbreviation**  $Disj A B \equiv A \wedge B \approx \perp$

**lemma** *Disj-comm*:  $\forall A B. Disj A B \longrightarrow Disj B A$  *<proof>*

**lemma** *Disj-IF*:  $\forall A. Disj (\mathcal{I} A) (\mathcal{F} A)$  *<proof>*

**lemma** *Disj-B*:  $\forall A. Disj (\mathcal{B} A) (\mathcal{B}(-A))$  *<proof>*

**lemma** *Disj-I*:  $\forall A. Disj (\mathcal{I} A) (-A)$  *<proof>*

**lemma** *Disj-BCI*:  $Fr-2 \mathcal{F} \implies \forall A. Disj (\mathcal{B}(C A)) (\mathcal{I}(-A))$  *<proof>*

**lemma** *Disj-CBI*:  $Fr-6 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A. Disj (C(\mathcal{B}(-A))) (\mathcal{I}(-A))$  *<proof>*

Introduces a predicate for indicating that two sets are separated and proves some properties.

**definition**  $Sep A B \equiv Disj (C A) B \wedge Disj (C B) A$

**lemma** *Sep-comm*:  $\forall A B. Sep A B \longrightarrow Sep B A$  *<proof>*

**lemma** *Sep-disj*:  $\forall A B. Sep A B \longrightarrow Disj A B$  *<proof>*

**lemma** *Sep-I*:  $Fr-1(\mathcal{F}) \implies Fr-2(\mathcal{F}) \implies Fr-4(\mathcal{F}) \implies \forall A. Sep (\mathcal{I} A) (\mathcal{I}(-A))$  *<proof>*

**lemma** *Sep-sub*:  $Fr-6 \mathcal{F} \implies \forall A B C D. Sep A B \wedge C \preceq A \wedge D \preceq B \longrightarrow Sep C D$  *<proof>*

**lemma** *Sep-Cl-diff*:  $Fr-6 \mathcal{F} \implies \forall A B. Cl(A) \wedge Cl(B) \longrightarrow Sep (A \leftarrow B) (B \leftarrow A)$  *<proof>*

**lemma** *Sep-Op-diff*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies \forall A B. Op(A) \wedge Op(B) \longrightarrow Sep (A \leftarrow B) (B \leftarrow A)$  *<proof>*

**lemma** *Sep-Cl*:  $\forall A B. Cl(A) \wedge Cl(B) \wedge Disj A B \longrightarrow Sep A B$  *<proof>*

**lemma** *Sep-Op*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies \forall A B. Op(A) \wedge Op(B) \wedge Disj A B \longrightarrow Sep A B$  *<proof>*

**lemma** *Fr-1a*:  $Fr-1a \mathcal{F} \implies Fr-2 \mathcal{F} \implies \forall A B C. Sep A B \wedge Sep A C \longrightarrow Sep A (B \vee C)$  *<proof>*

Verifies a neighborhood-based definition of closure.

**definition**  $nbhd A p \equiv \exists E. E \preceq A \wedge Op(E) \wedge (E p)$

**lemma** *nbhd-def2*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A p. (nbhd A p) = (\mathcal{I} A p)$  *<proof>*

**lemma** *C-def-lr*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A p. (C A p) \longrightarrow (\forall E. (nbhd E p) \longrightarrow \neg(Disj E A))$  *<proof>*

**lemma** *C-def-rl*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A p. (C A p) \longleftarrow (\forall E. (nbhd E p) \longrightarrow \neg(Disj E A))$  *<proof>*

**lemma** *C-def*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A p. (\mathcal{C} A p) \longleftrightarrow (\forall E. (nbhd E p) \longrightarrow \neg(Disj E A))$  *<proof>*

Explore the Barcan and converse Barcan formulas.

**lemma** *Barcan-I*:  $Fr-inf \mathcal{F} \implies \forall P. (\forall x. \mathcal{I}(P x)) \preceq \mathcal{I}(\forall x. P x)$  *<proof>*

**lemma** *Barcan-C*:  $Fr-2 \mathcal{F} \implies Fr-inf \mathcal{F} \implies \forall P. \mathcal{C}(\exists x. P x) \preceq (\exists x. \mathcal{C}(P x))$  *<proof>*

**lemma** *CBarcan-I*:  $Fr-1b \mathcal{F} \implies \forall P. \mathcal{I}(\forall x. P x) \preceq (\forall x. \mathcal{I}(P x))$  *<proof>*

**lemma** *CBarcan-C*:  $Fr-6 \mathcal{F} \implies \forall P. (\exists x. \mathcal{C}(P x)) \preceq \mathcal{C}(\exists x. P x)$  *<proof>*

**end**

**theory** *topo-negation-conditions*

**imports** *topo-frontier-algebra sse-operation-negative-quantification*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 7 Frontier-based negation - Semantic conditions

We define a paracomplete and a paraconsistent negation employing the interior and closure operation resp. We take the frontier operator as primitive and explore which semantic conditions are minimally required to obtain some relevant properties of negation.

**definition** *neg-I*:: $\sigma \Rightarrow \sigma$  ( $\neg^I$ ) **where**  $\neg^I A \equiv \mathcal{I}(\neg A)$

**definition** *neg-C*:: $\sigma \Rightarrow \sigma$  ( $\neg^C$ ) **where**  $\neg^C A \equiv \mathcal{C}(\neg A)$

**declare** *neg-I-def*[*conn*] *neg-C-def*[*conn*]

(We rename the meta-logical HOL negation to avoid terminological confusion)

**abbreviation** *cneg*::*bool* $\Rightarrow$ *bool* ( $\sim$ - [40]40) **where**  $\sim\varphi \equiv \neg\varphi$

### 7.1 'Explosion' (ECQ), non-contradiction (LNC) and excluded middle (TND)

TND

**lemma**  $\mathfrak{F} \mathcal{F} \implies TNDm \neg^I$  **nitpick** *<proof>*

**lemma** *TND-C*:  $TND \neg^C$  *<proof>*

ECQ

**lemma** *ECQ-I*:  $ECQ \neg^I$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies ECQm \neg^C$  **nitpick** *<proof>*

LNC

**lemma**  $LNC \neg^I$  **nitpick** *<proof>*

**lemma** *LNC-I*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies LNC \neg^I$  *<proof>*

**lemma**  $LNC \neg^C$  **nitpick** *<proof>*

**lemma** *LNC-C*:  $Fr-6 \mathcal{F} \implies LNC \neg^C$  *<proof>*

Relations between LNC and different ECQ variants (only relevant for paraconsistent negation).

**lemma**  $ECQ \neg^C \longrightarrow LNC \neg^C$  *<proof>*

**lemma** *ECQw-LNC*:  $ECQw \neg^C \longrightarrow LNC \neg^C$  *<proof>*

**lemma** *ECQm-LNC*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies ECQm \neg^C \longrightarrow LNC \neg^C$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies LNC \neg^C \longrightarrow ECQm \neg^C$  **nitpick** *<proof>*

Below we show that enforcing all conditions on the frontier operator still leaves room for both boldly paraconsistent and paracomplete models. We use Nitpick to generate a non-trivial model (here a set algebra with 8 elements).



**lemma**  $\mathfrak{F} \mathcal{F} \wedge \sim ECQm \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \wedge \sim TNDm \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

## 7.2 Modus tollens (MT)

MT-I

**lemma** *MT0-I*:  $Fr-1b \mathcal{F} \implies MT0 \neg^I$  *<proof>*

**lemma** *MT1-I*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies MT1 \neg^I$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies MT2 \neg^I$  **nitpick** *<proof>*

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge MT2 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma**  $\sim TNDm \neg^I \wedge Fr-1a \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge MT2 \neg^I$  **nitpick**[*satisfy*] *<proof>*

**lemma**  $\sim TNDm \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge MT2 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma**  $\sim TNDm \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge MT2 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies MT3 \neg^I$  **nitpick** *<proof>*

**lemma**  $\sim TNDm \neg^I \wedge Fr-1a \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge MT3 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma**  $\sim TNDm \neg^I \wedge MT0 \neg^I \wedge MT1 \neg^I \wedge MT2 \neg^I \wedge MT3 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

MT-C

**lemma**  $Fr-2 \mathcal{F} \implies MT0 \neg^C$  **nitpick** *<proof>*

**lemma** *MT0-C*:  $Fr-6 \mathcal{F} \implies MT0 \neg^C$  *<proof>*

**lemma** *MT1-C*:  $Fr-6 \mathcal{F} \implies MT1 \neg^C$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies MT2 \neg^C$  **nitpick** *<proof>*

**lemma**  $\sim ECQm \neg^C \wedge \mathfrak{F} \mathcal{F} \wedge MT2 \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma** *MT3-C*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies MT3 \neg^C$  *<proof>*

**lemma**  $\sim ECQm \neg^C \wedge MT0 \neg^C \wedge MT1 \neg^C \wedge MT2 \neg^C \wedge MT3 \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

## 7.3 Contraposition rules (CoP)

CoPw-I

**lemma**  $CoPw \neg^I$  **nitpick** *<proof>*

**lemma** *CoPw-I*:  $Fr-1b \mathcal{F} \implies CoPw \neg^I$  *<proof>*

CoPw-C

**lemma**  $CoPw \neg^C$  **nitpick** *<proof>*

**lemma** *CoPw-C*:  $Fr-6 \mathcal{F} \implies CoPw \neg^C$  *<proof>*

We can indeed prove that XCoP is entailed by CoP1 (CoP2) in the particular case of a closure-(interior-)based negation.

**lemma** *CoP1-XCoP*:  $CoP1 \neg^C \longrightarrow XCoP \neg^C$  *<proof>*

**lemma** *CoP2-XCoP*:  $CoP2 \neg^I \longrightarrow XCoP \neg^I$  *<proof>*

CoP1-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies CoP1 \neg^I$  **nitpick** *<proof>*

**lemma**  $\sim TNDm \neg^I \wedge \mathfrak{F} \mathcal{F} \wedge CoP1 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

CoP1-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies CoP1 \neg^C$  **nitpick** *<proof>*

**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge CoP1 \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

**lemma**  $CoP1 \neg^C \longrightarrow ECQm \neg^C$  *<proof>*

CoP2-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies CoP2 \neg^I$  **nitpick** *<proof>*

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge CoP2 \neg^I$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$   
**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge CoP2 \neg^I$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$   
**lemma**  $CoP2 \neg^I \longrightarrow TNDm \neg^I$   $\langle proof \rangle$

CoP2-C

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow CoP2 \neg^C$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim ECQm \neg^C \wedge \mathfrak{F} \mathcal{F} \wedge CoP2 \neg^C$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$

CoP3-I

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow CoP3 \neg^I$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim TND \neg^I \wedge CoP3 \neg^I$   $\langle proof \rangle$

CoP3-C

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow CoP3 \neg^C$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim ECQ \neg^C \wedge CoP3 \neg^C$   $\langle proof \rangle$

XCoP-I

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow XCoP \neg^I$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge XCoP \neg^I$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$   
**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge XCoP \neg^I$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$   
**lemma**  $XCoP \neg^I \longrightarrow TNDm \neg^I$   $\langle proof \rangle$

XCoP-C

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow XCoP \neg^C$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge XCoP \neg^C$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$   
**lemma**  $XCoP \neg^C \longrightarrow ECQm \neg^C$   $\langle proof \rangle$

## 7.4 Normality (negative) and its dual (nNor/nDNor)

nNor-I

**lemma**  $nNor \neg^I$  **nitpick**  $\langle proof \rangle$   
**lemma**  $nNor-I: Fr-2 \mathcal{F} \Longrightarrow Fr-3 \mathcal{F} \Longrightarrow nNor \neg^I$   $\langle proof \rangle$

nNor-C

**lemma**  $nNor-C: nNor \neg^C$   $\langle proof \rangle$

nDNor-I

**lemma**  $nDNor-I: nDNor \neg^I$   $\langle proof \rangle$

nDNor-C

**lemma**  $nDNor \neg^C$  **nitpick**  $\langle proof \rangle$   
**lemma**  $nDNor-C: Fr-3 \mathcal{F} \Longrightarrow nDNor \neg^C$   $\langle proof \rangle$

## 7.5 Double negation introduction/elimination (DNI/DNE)

DNI-I

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow DNI \neg^I$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim TNDm \neg^I \wedge \mathfrak{F} \mathcal{F} \wedge DNI \neg^I$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$

DNI-C

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow DNI \neg^C$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DNI \neg^C$  **nitpick**<sub>[satisfy, card w=3]</sub>  $\langle proof \rangle$

lemma  $\sim ECQm \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DNI \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim ECQm \neg^C \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DNI \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

DNE-I

lemma  $\mathfrak{F} \mathcal{F} \implies DNE \neg^I$  **nitpick** *<proof>*  
 lemma  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DNE \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DNE \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TNDm \neg^I \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DNE \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TND \neg^I \wedge DNE \neg^I \wedge DNI \neg^I$  *<proof>*

DNE-C

lemma  $\mathfrak{F} \mathcal{F} \implies DNE \neg^C$  **nitpick** *<proof>*  
 lemma  $\sim ECQm \neg^C \wedge \mathfrak{F} \mathcal{F} \wedge DNE \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

lemma  $\sim ECQ \neg^C \wedge DNE \neg^C \wedge DNI \neg^C$  *<proof>*

rDNI-I

lemma  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies rDNI \neg^I$  *<proof>*

rDNI-C

lemma  $Fr-3 \mathcal{F} \implies rDNI \neg^C$  *<proof>*

rDNE-I

lemma  $\mathfrak{F} \mathcal{F} \implies rDNE \neg^I$  **nitpick** *<proof>*  
 lemma  $\sim TNDm \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge rDNE \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TNDm \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge rDNE \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TNDm \neg^I \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge rDNE \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

rDNE-C

lemma  $\mathfrak{F} \mathcal{F} \implies rDNE \neg^C$  **nitpick** *<proof>*  
 lemma  $\sim ECQm \neg^C \wedge \mathfrak{F} \mathcal{F} \wedge rDNE \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

lemma  $\sim ECQm \neg^C \wedge rDNE \neg^C \wedge rDNI \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

## 7.6 De Morgan laws

DM1/2 (see CoPw)

DM3-I

lemma  $\mathfrak{F} \mathcal{F} \implies DM3 \neg^I$  **nitpick** *<proof>*  
 lemma  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DM3 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DM3 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*  
 lemma  $\sim TNDm \neg^I \wedge DM3 \neg^I$  *<proof>*

DM3-C

lemma  $DM3 \neg^C$  **nitpick** *<proof>*  
 lemma  $DM3-C: Fr-1a \mathcal{F} \implies Fr-2 \mathcal{F} \implies DM3 \neg^C$  *<proof>*  
 lemma  $\sim ECQm \neg^C \wedge \mathfrak{F} \mathcal{F} \wedge DM3 \neg^C$  **nitpick**[*satisfy, card w=3*] *<proof>*

DM4-I

lemma  $DM4 \neg^I$  **nitpick** *<proof>*  
 lemma  $DM4-I: Fr-1a \mathcal{F} \implies DM4 \neg^I$  *<proof>*  
 lemma  $\sim TNDm \neg^I \wedge \mathfrak{F} \mathcal{F} \wedge DM4 \neg^I$  **nitpick**[*satisfy, card w=3*] *<proof>*

DM4-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies DM_4 \neg^C$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge DM_4 \neg^C$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $\sim ECQm \neg^C \wedge DM_4 \neg^C$   $\langle proof \rangle$

iDM1/2 (see CoPw)

iDM3-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies Fr-inf \mathcal{F} \implies iDM3 \neg^I$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge iDM3 \neg^I$  **nitpick** $[satisfy]$   $\langle proof \rangle$

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge iDM3 \neg^I$  **nitpick** $[satisfy]$   $\langle proof \rangle$

**lemma**  $\sim TNDm \neg^I \wedge iDM3 \neg^I$   $\langle proof \rangle$

iDM3-C

**lemma**  $iDM3 \neg^C$  **nitpick**  $\langle proof \rangle$

**lemma**  $iDM3-C: Fr-2 \mathcal{F} \implies Fr-inf \mathcal{F} \implies iDM3 \neg^C$   $\langle proof \rangle$

iDM4-I

**lemma**  $iDM_4 \neg^I$  **nitpick**  $\langle proof \rangle$

**lemma**  $iDM_4-I: Fr-inf \mathcal{F} \implies iDM_4 \neg^I$   $\langle proof \rangle$

iDM4-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies Fr-inf \mathcal{F} \implies iDM_4 \neg^C$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge iDM_4 \neg^C$  **nitpick** $[satisfy]$   $\langle proof \rangle$

**lemma**  $\sim ECQm \neg^C \wedge iDM_4 \neg^C$   $\langle proof \rangle$

## 7.7 Local contraposition axioms (lCoP)

lCoPw-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoPw(\rightarrow) \neg^I$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge lCoPw(\rightarrow) \neg^I$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge lCoPw(\rightarrow) \neg^I$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $lCoPw(\rightarrow) \neg^I \longrightarrow TNDm \neg^I$   $\langle proof \rangle$

lCoPw-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoPw(\rightarrow) \neg^C$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge lCoPw(\rightarrow) \neg^C$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $lCoPw(\rightarrow) \neg^C \longrightarrow ECQm \neg^C$   $\langle proof \rangle$

lCoP1-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP1(\rightarrow) \neg^I$  **nitpick**  $\langle proof \rangle$

**lemma**  $lCoP1(\rightarrow) \neg^I \longrightarrow TND \neg^I$   $\langle proof \rangle$

lCoP1-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP1(\rightarrow) \neg^C$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim ECQ \neg^C \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge lCoP1(\rightarrow) \neg^C$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $lCoP1(\rightarrow) \neg^C \longrightarrow ECQm \neg^C$   $\langle proof \rangle$

lCoP2-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP2(\rightarrow) \neg^I$  **nitpick**  $\langle proof \rangle$

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge lCoP2(\rightarrow) \neg^I$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $\sim TND \neg^I \wedge Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge lCoP2(\rightarrow) \neg^I$  **nitpick** $[satisfy, card w=3]$   $\langle proof \rangle$

**lemma**  $lCoP2(\rightarrow) \neg^I \rightarrow TNDm \neg^I \langle proof \rangle$

lCoP2-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP2(\rightarrow) \neg^C \text{nitpick} \langle proof \rangle$

**lemma**  $lCoP2(\rightarrow) \neg^C \rightarrow ECQ \neg^C \langle proof \rangle$

lCoP3-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP3(\rightarrow) \neg^I \text{nitpick} \langle proof \rangle$

**lemma**  $lCoP3(\rightarrow) \neg^I \rightarrow TND \neg^I \langle proof \rangle$

lCoP3-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP3(\rightarrow) \neg^C \text{nitpick} \langle proof \rangle$

**lemma**  $lCoP3(\rightarrow) \neg^C \rightarrow ECQ \neg^C \langle proof \rangle$

## 7.8 Disjunctive syllogism

DS1-I

**lemma**  $DS1(\rightarrow) \neg^I \langle proof \rangle$

DS1-C

**lemma**  $\mathfrak{F} \mathcal{F} \implies DS1(\rightarrow) \neg^C \text{nitpick} \langle proof \rangle$

**lemma**  $DS1(\rightarrow) \neg^C \rightarrow ECQ \neg^C \langle proof \rangle$

DS2-I

**lemma**  $\mathfrak{F} \mathcal{F} \implies DS2(\rightarrow) \neg^I \text{nitpick} \langle proof \rangle$

**lemma**  $DS2(\rightarrow) \neg^I \rightarrow TND \neg^I \langle proof \rangle$

DS2-C

**lemma**  $DS2(\rightarrow) \neg^C \langle proof \rangle$

**end**

**theory** *topo-negation-fixedpoints*

**imports** *topo-frontier-algebra sse-operation-negative-quantification*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 8 Frontier-based negation - Fixed-points

We define a paracomplete and a paraconsistent negation employing the interior and the closure operation respectively. We explore the use of fixed-point predicates to recover some relevant properties of negation, many of which cannot be readily recovered by just adding semantic conditions. We take the frontier operator as primitive and explore which semantic conditions are minimally required.

**definition**  $neg-I::\sigma \Rightarrow \sigma \ (\neg^I) \text{ where } \neg^I \varphi \equiv \mathcal{I}(-\varphi)$

**definition**  $neg-C::\sigma \Rightarrow \sigma \ (\neg^C) \text{ where } \neg^C \varphi \equiv \mathcal{C}(-\varphi)$

**declare**  $neg-I-def[conn] \ neg-C-def[conn]$

Note that all results obtained for fixed-point predicates extend to their associated operators as follows:

**lemma**  $\forall A. \gamma^{fp}(A) \wedge \varphi(A) \preceq \psi(A) \implies \forall A. (fp \ \gamma)(A) \rightarrow \varphi(A) \preceq \psi(A) \langle proof \rangle$

**lemma**  $\forall A B. \gamma^{fp}(A) \wedge \gamma^{fp}(B) \wedge (\varphi A B) \preceq (\psi A B) \implies \forall A B. (fp \ \gamma)(A) \wedge (fp \ \gamma)(B) \longrightarrow (\varphi A B) \preceq (\psi A B)$  *<proof>*

Recall from previous results that if we have  $Fr(A)$  then we also have both  $Cl(A)$  and  $Br(A)$ . With this understanding we will tacitly assume the corresponding results for  $Fr(-)$  below. Moreover, we obtained countermodels (using Nitpick) for all formulas featuring other combinations (not shown).

## 8.1 'Explosion' (ECQ) and excluded middle (TND)

TND-I

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A. Cl(A) \longrightarrow TND^A \neg^I$  *<proof>*

ECQ-C

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A. Op(A) \longrightarrow ECQ^A \neg^C$  *<proof>*

## 8.2 Contraposition rules

CoPw-I

**lemma**  $\forall A B. Br(-B) \longrightarrow CoPw^{AB} \neg^I$  *<proof>*

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow CoPw^{AB} \neg^I$  *<proof>*

CoPw-C

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Br(A) \longrightarrow CoPw^{AB} \neg^C$  *<proof>*

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Op(B) \longrightarrow CoPw^{AB} \neg^C$  *<proof>*

CoP1-I

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow CoP1^{AB} \neg^I$  *<proof>*

**lemma**  $Fr-1b \ \mathcal{F} \implies \forall A B. Op(B) \longrightarrow CoP1^{AB} \neg^I$  *<proof>*

**lemma**  $CoP1-I-rec: Fr-2 \ \mathcal{F} \implies Fr-3 \ \mathcal{F} \implies \forall A B. Br(-B) \longrightarrow CoP1^{AB} \neg^I$  *<proof>*

CoP1-C

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Op(B) \longrightarrow CoP1^{AB} \neg^C$  *<proof>*

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Br(A) \longrightarrow CoP1^{AB} \neg^C$  *<proof>*

CoP2-I

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow CoP2^{AB} \neg^I$  *<proof>*

**lemma**  $\forall A B. Br(-B) \longrightarrow CoP2^{AB} \neg^I$  *<proof>*

CoP2-C

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Op(B) \longrightarrow CoP2^{AB} \neg^C$  *<proof>*

**lemma**  $Fr-6 \ \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow CoP2^{AB} \neg^C$  *<proof>*

**lemma**  $Fr-2 \ \mathcal{F} \implies Fr-3 \ \mathcal{F} \implies \forall A B. Br(A) \longrightarrow CoP2^{AB} \neg^C$  *<proof>*

CoP3-I

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow CoP3^{AB} \neg^I$  *<proof>*

CoP3-C

**lemma**  $Fr-2 \ \mathcal{F} \implies \forall A B. Op(B) \longrightarrow CoP3^{AB} \neg^C$  *<proof>*

XCoP-I

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Cl(A) \longrightarrow XCoP^{AB} \neg^I \langle proof \rangle$

**lemma**  $\forall A B. Br(-B) \longrightarrow XCoP^{AB} \neg^I \langle proof \rangle$

XCoP-C

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Op(B) \longrightarrow XCoP^{AB} \neg^C \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. \forall A B. Br(A) \longrightarrow XCoP^{AB} \neg^C \langle proof \rangle$

### 8.3 Double negation introduction/elimination

DNI-I

**lemma** *Fr-1b*  $\mathcal{F} \implies \forall A. Op(A) \longrightarrow DNI^A \neg^I \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall A. Br(-A) \longrightarrow DNI^A \neg^I \langle proof \rangle$

DNI-C

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A. Op(A) \longrightarrow DNI^A \neg^C \langle proof \rangle$

DNE-I

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A. Cl(A) \longrightarrow DNE^A \neg^I \langle proof \rangle$

DNE-C

**lemma** *Fr-6*  $\mathcal{F} \implies \forall A. Cl(A) \longrightarrow DNE^A \neg^C \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall A. Br(A) \longrightarrow DNE^A \neg^C \langle proof \rangle$

### 8.4 De Morgan laws

DM1-I

**lemma** *Fr-1b*  $\mathcal{F} \implies \forall A B. DM1^{AB} \neg^I \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Cl(A) \wedge Cl(B) \longrightarrow DM1^{AB} \neg^I \langle proof \rangle$

DM1-C

**lemma** *Fr-6*  $\mathcal{F} \implies \forall A B. DM1^{AB} \neg^C \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Br(A) \wedge Br(B) \longrightarrow DM1^{AB} \neg^C \langle proof \rangle$

DM2-I

**lemma** *Fr-1b*  $\mathcal{F} \implies \forall A B. DM2^{AB} \neg^I \langle proof \rangle$

**lemma**  $\forall A B. Br(-A) \wedge Br(-B) \longrightarrow DM2^{AB} \neg^I \langle proof \rangle$

DM2-C

**lemma** *Fr-6*  $\mathcal{F} \implies \forall A B. DM2^{AB} \neg^C \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Op(A) \wedge Op(B) \longrightarrow DM2^{AB} \neg^C \langle proof \rangle$

DM3-I

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Cl(A) \wedge Cl(B) \longrightarrow DM3^{AB} \neg^I \langle proof \rangle$

DM3-C

**lemma** *Fr-1a*  $\mathcal{F} \implies Fr-2 \mathcal{F} \implies \forall A B. DM3^{AB} \neg^C \langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Br(A) \wedge Br(B) \longrightarrow DM3^{AB} \neg^C \langle proof \rangle$

DM4-I

**lemma** *Fr-1a*  $\mathcal{F} \implies Fr-2 \mathcal{F} \implies \forall A B. DM4^{AB} \neg^I \langle proof \rangle$

**lemma**  $\forall A B. Br(-A) \wedge Br(-B) \longrightarrow DM4^{AB} \neg^I \langle proof \rangle$

DM4-C

**lemma** *Fr-2*  $\mathcal{F} \implies \forall A B. Op(A) \wedge Op(B) \longrightarrow DM4^{AB} \neg^C \langle proof \rangle$

**lemma** *Fr-1*  $\mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall A B. Fr(A) \wedge Fr(B) \longrightarrow DM4^{AB} \neg^C \langle proof \rangle$

## 8.5 Local contraposition axioms

lCoPw-I

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow lCoPw^{AB}(\rightarrow) \neg^I \langle proof \rangle$

**lemma**  $\forall A B. Br(-B) \longrightarrow lCoPw^{AB}(\rightarrow) \neg^I \langle proof \rangle$

lCoPw-C

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Op(B) \longrightarrow lCoPw^{AB}(\rightarrow) \neg^C \langle proof \rangle$

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Br(A) \longrightarrow lCoPw^{AB}(\rightarrow) \neg^C \langle proof \rangle$

lCoP1-I

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow lCoP1^{AB}(\rightarrow) \neg^I \langle proof \rangle$

lCoP1-C

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Op(B) \longrightarrow lCoP1^{AB}(\rightarrow) \neg^C \langle proof \rangle$

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Br(A) \longrightarrow lCoP1^{AB}(\rightarrow) \neg^C \langle proof \rangle$

lCoP2-I

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow lCoP2^{AB}(\rightarrow) \neg^I \langle proof \rangle$

**lemma**  $\forall A B. Br(-B) \longrightarrow lCoP2^{AB}(\rightarrow) \neg^I \langle proof \rangle$

lCoP2-C

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Op(B) \longrightarrow lCoP2^{AB}(\rightarrow) \neg^C \langle proof \rangle$

lCoP3-I

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow lCoP3^{AB}(\rightarrow) \neg^I \langle proof \rangle$

lCoP3-C

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Op(B) \longrightarrow lCoP3^{AB}(\rightarrow) \neg^C \langle proof \rangle$

## 8.6 Disjunctive syllogism

DS1-I

**lemma**  $\forall A B. DS1^{AB}(\rightarrow) \neg^I \langle proof \rangle$

DS1-C

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Op(A) \longrightarrow DS1^{AB}(\rightarrow) \neg^C \langle proof \rangle$

DS2-I

**lemma**  $Fr-2 \mathcal{F} \implies \forall A B. Cl(A) \longrightarrow DS2^{AB}(\rightarrow) \neg^I \langle proof \rangle$

DS2-C

**lemma**  $\forall A B. DS2^{AB}(\rightarrow) \neg^C \langle proof \rangle$

**end**

**theory** *ex-LFIs*

**imports** *topo-negation-conditions*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]



## 9 Example application: Logics of Formal Inconsistency (LFIs)

The LFIs [4] [3] are a family of paraconsistent logics featuring a 'consistency' operator  $\circ$  that can be used to recover some classical properties of negation (in particular ECQ). We show how to semantically embed LFIs as extensions of Boolean algebras (here as frontier algebras).

Logical validity can be defined as truth in all worlds/points (i.e. as denoting the top element)

**abbreviation**  $gtrue::\sigma \Rightarrow bool$  ( $[\vdash -]$ ) **where**  $[\vdash A] \equiv \forall w. A w$

**lemma**  $gtrue-def: [\vdash A] \equiv A \approx \top$  *<proof>*

When defining a logic over an existing algebra we have two choices: a global (truth preserving) and a local (truth-degree preserving) notion of logical consequence. For LFIs we prefer the latter.

**abbreviation**  $conseq-global1::\sigma \Rightarrow \sigma \Rightarrow bool$  ( $[- \vdash_g -]$ ) **where**  $[A \vdash_g B] \equiv [\vdash A] \longrightarrow [\vdash B]$

**abbreviation**  $conseq-global2::\sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow bool$  ( $[-, - \vdash_g -]$ ) **where**  $[A_1, A_2 \vdash_g B] \equiv [\vdash A_1] \wedge [\vdash A_2] \longrightarrow [\vdash B]$

**abbreviation**  $conseq-global3::\sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow bool$  ( $[-, -, - \vdash_g -]$ ) **where**  $[A_1, A_2, A_3 \vdash_g B] \equiv [\vdash A_1] \wedge [\vdash A_2] \wedge [\vdash A_3] \longrightarrow [\vdash B]$

**abbreviation**  $conseq-local1::\sigma \Rightarrow \sigma \Rightarrow bool$  ( $[- \vdash -]$ ) **where**  $[A \vdash B] \equiv A \preceq B$

**abbreviation**  $conseq-local2::\sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow bool$  ( $[-, - \vdash -]$ ) **where**  $[A_1, A_2 \vdash B] \equiv A_1 \wedge A_2 \preceq B$

**abbreviation**  $conseq-local3::\sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow bool$  ( $[-, -, - \vdash -]$ ) **where**  $[A_1, A_2, A_3 \vdash B] \equiv A_1 \wedge A_2 \wedge A_3 \preceq B$

For LFIs we use the (paraconsistent) closure-based negation previously defined (taking frontier as primitive).

**abbreviation**  $cneg::\sigma \Rightarrow \sigma$  ( $\neg$ ) **where**  $\neg A \equiv \neg^C A$

In terms of the frontier operator the negation looks as follows:

**lemma**  $\neg A \approx -A \vee \mathcal{F}(-A)$  *<proof>*

**lemma**  $cneg-prop: Fr-2 \mathcal{F} \Longrightarrow \neg A \approx -A \vee \mathcal{F}(A)$  *<proof>*

This negation is of course boldly paraconsistent.

**lemma**  $[a, \neg a \vdash b]$  **nitpick** *<proof>*

**lemma**  $[a, \neg a \vdash \neg b]$  **nitpick** *<proof>*

**lemma**  $[a, \neg a \vdash_g b]$  **nitpick** *<proof>*

**lemma**  $[a, \neg a \vdash_g \neg b]$  **nitpick** *<proof>*

We define two pairs of in/consistency operators and show how they relate to each other. Using LFIs terminology, the minimal logic so encoded corresponds to 'RmbC-ciw' (cf. [3]).

**abbreviation**  $op-inc-a::\sigma \Rightarrow \sigma$  ( $\cdot^A$ - [57]58) **where**  $\cdot^A A \equiv A \wedge \neg A$

**abbreviation**  $op-con-a::\sigma \Rightarrow \sigma$  ( $\circ^A$ - [57]58) **where**  $\circ^A A \equiv -\cdot^A A$

**abbreviation**  $op-inc-b::\sigma \Rightarrow \sigma$  ( $\cdot^B$ - [57]58) **where**  $\cdot^B A \equiv \mathcal{B} A$

**abbreviation**  $op-con-b::\sigma \Rightarrow \sigma$  ( $\circ^B$ - [57]58) **where**  $\circ^B A \equiv \mathcal{B}^c A$

Observe that assuming condition Fr-2 are we allowed to exchange A and B variants.

**lemma**  $pincAB: Fr-2 \mathcal{F} \Longrightarrow \cdot^A A \approx \cdot^B A$  *<proof>*

**lemma**  $pconAB: Fr-2 \mathcal{F} \Longrightarrow \circ^A A \approx \circ^B A$  *<proof>*

Variants A and B give us slightly different properties.

**lemma**  $Prop1: \circ^B A \approx \mathcal{I}^{fp} A$  *<proof>*

**lemma**  $\circ^A A \approx A \rightarrow \mathcal{I} A$  **nitpick** *<proof>*

**lemma**  $Prop2: Cl A \longleftrightarrow \circ^A -A \approx \top$  *<proof>*

**lemma**  $Cl\ A \longrightarrow \circ^B - A \approx \top$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Prop3: Cl\ A \longleftrightarrow \cdot^A - A \approx \perp$   $\langle proof \rangle$   
**lemma**  $Cl\ A \longrightarrow \cdot^B - A \approx \perp$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Prop4: Op\ A \longleftrightarrow \circ^B A \approx \top$   $\langle proof \rangle$   
**lemma**  $Op\ A \longrightarrow \circ^A A \approx \top$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Prop5: Op\ A \longleftrightarrow \cdot^B A \approx \perp$   $\langle proof \rangle$   
**lemma**  $Op\ A \longrightarrow \cdot^A A \approx \perp$  **nitpick**  $\langle proof \rangle$

Importantly, LFIs must satisfy the so-called 'principle of gentle explosion'. Only variant A works here:

**lemma**  $[\circ^A a, a, \neg a \vdash b]$   $\langle proof \rangle$   
**lemma**  $[\circ^A a, a, \neg a \vdash_g b]$   $\langle proof \rangle$   
**lemma**  $[\circ^B a, a, \neg a \vdash b]$  **nitpick**  $\langle proof \rangle$   
**lemma**  $[\circ^B a, a, \neg a \vdash_g b]$  **nitpick**  $\langle proof \rangle$

In what follows we employ the (minimal) A-variant and verify some properties.

**abbreviation**  $op-inc :: \sigma \Rightarrow \sigma (\cdot - [57]58)$  **where**  $\cdot A \equiv \cdot^A A$   
**abbreviation**  $op-con :: \sigma \Rightarrow \sigma (\circ - [57]58)$  **where**  $\circ A \equiv \neg \cdot A$

**lemma**  $TND(\neg)$   $\langle proof \rangle$   
**lemma**  $ECQm(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Fr-1b\ \mathcal{F} \Longrightarrow Fr-2\ \mathcal{F} \Longrightarrow LNC(\neg)$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow DNI(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow DNE(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Fr-1b\ \mathcal{F} \Longrightarrow Fr-2\ \mathcal{F} \Longrightarrow CoPw(\neg)$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow CoP1(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow CoP2(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow CoP3(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Fr-1a\ \mathcal{F} \Longrightarrow Fr-2\ \mathcal{F} \Longrightarrow DM3(\neg)$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow DM4(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $nNor(\neg)$   $\langle proof \rangle$   
**lemma**  $Fr-3\ \mathcal{F} \Longrightarrow nDNor(\neg)$   $\langle proof \rangle$   
**lemma**  $Fr-1b\ \mathcal{F} \Longrightarrow Fr-2\ \mathcal{F} \Longrightarrow MT0(\neg)$   $\langle proof \rangle$   
**lemma**  $Fr-1b\ \mathcal{F} \Longrightarrow Fr-2\ \mathcal{F} \Longrightarrow MT1(\neg)$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow MT2(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Fr-1b\ \mathcal{F} \Longrightarrow Fr-2\ \mathcal{F} \Longrightarrow Fr-3\ \mathcal{F} \Longrightarrow MT3(\neg)$   $\langle proof \rangle$

We show how all local contraposition variants (lCoP) can be recovered using the consistency operator. Observe that we can recover in the same way other (weaker) properties: CoP, MT and DNI/DNE (local & global).

**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow lCoPw(\rightarrow)(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $cons-lcop1: [\circ b, a \rightarrow b \vdash \neg b \rightarrow \neg a]$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow lCoP1(\rightarrow)(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $cons-lcop2: [\circ b, a \rightarrow \neg b \vdash b \rightarrow \neg a]$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow lCoP2(\rightarrow)(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $cons-lcop3: [\circ b, \neg a \rightarrow b \vdash \neg b \rightarrow a]$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow lCoP3(\rightarrow)(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $cons-lcop4: [\circ b, \neg a \rightarrow \neg b \vdash b \rightarrow a]$   $\langle proof \rangle$

Disjunctive syllogism (DS).

**lemma**  $\mathfrak{F}\ \mathcal{F} \Longrightarrow DS1(\rightarrow)(\neg)$  **nitpick**  $\langle proof \rangle$   
**lemma**  $cons-ds1: [\circ a, a \vee b \vdash \neg a \rightarrow b]$   $\langle proof \rangle$   
**lemma**  $DS2(\rightarrow)(\neg)$   $\langle proof \rangle$

Further properties.

**lemma**  $[a \wedge \neg a \vdash \neg(\circ a)]$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies [\neg(\circ a) \vdash a \wedge \neg a]$  **nitpick**  $\langle proof \rangle$   
**lemma**  $[\circ a \vdash \neg(a \wedge \neg a)]$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies [\neg(a \wedge \neg a) \vdash \circ a]$  **nitpick**  $\langle proof \rangle$

The following axioms are commonly employed in the literature on LFIs to obtain stronger logics. We explore under which conditions they can be assumed while keeping the logic boldly paraconsistent.

**abbreviation** *cf* **where**  $cf \equiv DNE(\neg)$   
**abbreviation** *ce* **where**  $ce \equiv DNI(\neg)$   
**abbreviation** *ciw* **where**  $ciw \equiv \forall P. [\vdash \circ P \vee \cdot P]$   
**abbreviation** *ci* **where**  $ci \equiv \forall P. [\neg(\circ P) \vdash \cdot P]$   
**abbreviation** *cl* **where**  $cl \equiv \forall P. [\neg(\cdot P) \vdash \circ P]$   
**abbreviation** *ca-conj* **where**  $ca-conj \equiv \forall P Q. [\circ P, \circ Q \vdash \circ(P \wedge Q)]$   
**abbreviation** *ca-disj* **where**  $ca-disj \equiv \forall P Q. [\circ P, \circ Q \vdash \circ(P \vee Q)]$   
**abbreviation** *ca-impl* **where**  $ca-impl \equiv \forall P Q. [\circ P, \circ Q \vdash \circ(P \rightarrow Q)]$   
**abbreviation** *ca* **where**  $ca \equiv ca-conj \wedge ca-disj \wedge ca-impl$

*cf*

**lemma**  $\mathfrak{F} \mathcal{F} \implies cf$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \wedge cf \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$

*ce*

**lemma**  $\mathfrak{F} \mathcal{F} \implies ce$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge ce \wedge \sim ECQ(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$   
**lemma**  $Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge ce \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$   
**lemma**  $Fr-1a \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge ce \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$   
**lemma**  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies ce \longrightarrow ECQm(\neg)$   $\langle proof \rangle$

*ciw*

**lemma** *ciw*  $\langle proof \rangle$

*ci*

**lemma**  $\mathfrak{F} \mathcal{F} \implies ci$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \wedge ci \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$

*cl*

**lemma**  $\mathfrak{F} \mathcal{F} \implies cl$  **nitpick**  $\langle proof \rangle$   
**lemma**  $Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge cl \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$   
**lemma**  $Fr-1a \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge cl \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$   
**lemma**  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies cl \longrightarrow ECQ(\neg)$   $\langle proof \rangle$

*ca-conj/disj*

**lemma**  $Fr-1a \mathcal{F} \implies Fr-2 \mathcal{F} \implies ca-conj$   $\langle proof \rangle$   
**lemma**  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies ca-disj$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies ca-impl$  **nitpick**  $\langle proof \rangle$

*ca-impl*

**lemma**  $ca-impl \wedge \sim ECQ(\neg)$   $\langle proof \rangle$   
**lemma**  $ca-impl \longrightarrow ECQm(\neg)$   $\langle proof \rangle$

*cf & ci*

**lemma**  $Fr-1 \mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge cf \wedge ci \wedge \sim ECQm(\neg)$  **nitpick** $[satisfy]$   $\langle proof \rangle$

**lemma** *Fr-2*  $\mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge cf \wedge ci \wedge \sim ECQm(\neg)$  **nitpick**[*satisfy*]  $\langle proof \rangle$   
**lemma** *Fr-1b*  $\mathcal{F} \wedge Fr-2 \mathcal{F} \wedge cf \wedge ci \wedge \sim ECQ(\neg)$   $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \wedge cf \wedge ci \longrightarrow ECQm(\neg)$   $\langle proof \rangle$

cf & cl

**lemma** *Fr-1*  $\mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge cf \wedge cl \wedge \sim ECQm(\neg)$  **nitpick**[*satisfy*]  $\langle proof \rangle$   
**lemma** *Fr-2*  $\mathcal{F} \wedge Fr-3 \mathcal{F} \wedge Fr-4 \mathcal{F} \wedge cf \wedge cl \wedge \sim ECQm(\neg)$  **nitpick**[*satisfy*]  $\langle proof \rangle$   
**lemma** *Fr-1b*  $\mathcal{F} \wedge Fr-2 \mathcal{F} \wedge cf \wedge cl \longrightarrow ECQ(\neg)$   $\langle proof \rangle$

**end**

**theory** *topo-strict-implication*

**imports** *topo-frontier-algebra*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 10 Strict implication

We can also employ the closure and interior operations to define different sorts of implications. In this section we preliminary explore a sort of strict implication and check some relevant properties.

A 'strict' implication should entail the classical one. Hence we define it using the interior operator.

**definition** *imp-I*:  $\sigma \Rightarrow \sigma \Rightarrow \sigma$  (**infix**  $\Rightarrow$  51) **where**  $\alpha \Rightarrow \beta \equiv \mathcal{I}(\alpha \rightarrow \beta)$   
**abbreviation** *imp-I'*:  $\sigma \Rightarrow \sigma \Rightarrow \sigma$  (**infix**  $\Leftarrow$  51) **where**  $\beta \Leftarrow \alpha \equiv \alpha \Rightarrow \beta$   
**declare** *imp-I-def*[*conn*]

**lemma** *imp-rel*:  $\forall a b. a \Rightarrow b \preceq a \rightarrow b$   $\langle proof \rangle$

Under appropriate conditions this implication satisfies a weak variant of the deduction theorem (DT),

**lemma** *DTw1*:  $\forall a b. a \Rightarrow b \approx \top \longrightarrow a \preceq b$   $\langle proof \rangle$   
**lemma** *DTw2*: *Fr-2*  $\mathcal{F} \Longrightarrow Fr-3 \mathcal{F} \Longrightarrow \forall a b. a \preceq b \longrightarrow a \Rightarrow b \approx \top$   $\langle proof \rangle$

and a variant of modus ponens and modus tollens resp.

**lemma** *MP*:  $\forall a b. a \wedge (a \Rightarrow b) \preceq b$   $\langle proof \rangle$   
**lemma** *MT*:  $\forall a b. (a \Rightarrow b) \wedge \neg b \preceq \neg a$   $\langle proof \rangle$

However the full DT (actually right-to-left: implication introduction) is not valid.

**lemma** *DT1*:  $\forall a b X. X \preceq a \Rightarrow b \longrightarrow X \wedge a \preceq b$   $\langle proof \rangle$   
**lemma** *DT2*:  $\mathfrak{F} \mathcal{F} \Longrightarrow \forall a b X. X \wedge a \preceq b \longrightarrow X \preceq a \Rightarrow b$  **nitpick**  $\langle proof \rangle$

This implication has thus a sort of 'relevant' behaviour. For instance, the following are not valid:

**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow \forall a b. (a \Rightarrow (b \Rightarrow a)) \approx \top$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow \forall a b. (a \Rightarrow ((a \Rightarrow b) \Rightarrow b)) \approx \top$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow \forall a b c. (a \Rightarrow b) \vee (b \Rightarrow c) \approx \top$  **nitpick**  $\langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \Longrightarrow \forall a b. ((a \Rightarrow b) \Rightarrow a) \Rightarrow a \approx \top$  **nitpick**  $\langle proof \rangle$

In contrast the properties below are valid for appropriate conditions.

**lemma** *iprop0*: *Fr-2*  $\mathcal{F} \Longrightarrow Fr-3 \mathcal{F} \Longrightarrow \forall a. a \Rightarrow a \approx \top$   $\langle proof \rangle$   
**lemma** *iprop1*: *Fr-2*  $\mathcal{F} \Longrightarrow Fr-3 \mathcal{F} \Longrightarrow \forall a b. a \wedge (a \Rightarrow b) \Rightarrow b \approx \top$   $\langle proof \rangle$

**lemma** *iprop2*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. a \Rightarrow (b \Rightarrow b) \approx \top$  *<proof>*

**lemma** *iprop3*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. ((a \Rightarrow a) \Rightarrow b) \Rightarrow b \approx \top$  *<proof>*

**lemma** *iprop4*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. (a \wedge b) \Rightarrow a \approx \top$  *<proof>*

**lemma** *iprop5*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. a \Rightarrow (a \vee b) \approx \top$  *<proof>*

**lemma** *iprop6a*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. (a \wedge (b \vee c)) \Rightarrow ((a \wedge b) \vee (a \wedge c)) \approx \top$  *<proof>*

**lemma** *iprop6b*:  $Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. (a \wedge (b \vee c)) \Leftarrow ((a \wedge b) \vee (a \wedge c)) \approx \top$  *<proof>*

**lemma** *iprop7'*:  $Fr-1 \mathcal{F} \implies \forall a b c. (a \Rightarrow b) \wedge (b \Rightarrow c) \preceq (a \Rightarrow c)$  *<proof>*

**lemma** *iprop7*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (b \Rightarrow c)) \Rightarrow (a \Rightarrow c) \approx \top$  *<proof>*

**lemma** *iprop8a'*:  $Fr-1 \mathcal{F} \implies \forall a b c. (a \Rightarrow b) \wedge (a \Rightarrow c) \preceq a \Rightarrow (b \wedge c)$  *<proof>*

**lemma** *iprop8b'*:  $Fr-1b \mathcal{F} \implies \forall a b c. (a \Rightarrow b) \wedge (a \Rightarrow c) \succeq a \Rightarrow (b \wedge c)$  *<proof>*

**lemma** *iprop8a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (a \Rightarrow c)) \Rightarrow (a \Rightarrow (b \wedge c)) \approx \top$  *<proof>*

**lemma** *iprop8b*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (a \Rightarrow c)) \Leftarrow (a \Rightarrow (b \wedge c)) \approx \top$  *<proof>*

**lemma** *iprop9a'*:  $Fr-1 \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (c \Rightarrow b)) \preceq ((a \vee c) \Rightarrow b)$  *<proof>*

**lemma** *iprop9b'*:  $Fr-1b \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (c \Rightarrow b)) \succeq ((a \vee c) \Rightarrow b)$  *<proof>*

**lemma** *iprop9a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (c \Rightarrow b)) \Rightarrow ((a \vee c) \Rightarrow b) \approx \top$  *<proof>*

**lemma** *iprop9b*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. ((a \Rightarrow b) \wedge (c \Rightarrow b)) \Leftarrow ((a \vee c) \Rightarrow b) \approx \top$  *<proof>*

**lemma** *iprop10'*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall a b c. a \Rightarrow (b \Rightarrow c) \preceq (a \Rightarrow b) \Rightarrow (a \Rightarrow c)$  *<proof>*

**lemma** *iprop10*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall a b c. (a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c)) \approx \top$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b c. a \Rightarrow (b \Rightarrow c) \succeq (a \Rightarrow b) \Rightarrow (a \Rightarrow c)$  **nitpick** *<proof>*

**lemma** *iprop11a'*:  $Fr-1 \mathcal{F} \implies \forall a b. (a \Rightarrow (a \Rightarrow b)) \preceq (a \Rightarrow b)$  *<proof>*

**lemma** *iprop11b'*:  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \Rightarrow (a \Rightarrow b)) \succeq (a \Rightarrow b)$  *<proof>*

**lemma** *iprop11a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. (a \Rightarrow (a \Rightarrow b)) \Rightarrow (a \Rightarrow b) \approx \top$  *<proof>*

**lemma** *iprop11b*:  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \Rightarrow (a \Rightarrow b)) \Leftarrow (a \Rightarrow b) \approx \top$  *<proof>*

In particular, 'strengthening the antecedent' is valid only under certain conditions:

**lemma** *SA'*:  $Fr-1b \mathcal{F} \implies \forall a b c. a \Rightarrow c \preceq (a \wedge b) \Rightarrow c$  *<proof>*

**lemma** *SA*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. (a \Rightarrow c) \Rightarrow ((a \wedge b) \Rightarrow c) \approx \top$  *<proof>*

**lemma** *Fr-1a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall a b c. a \Rightarrow c \preceq (a \wedge b) \Rightarrow c$  **nitpick** *<proof>*

**lemma** *Fr-1a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall a b c. (a \Rightarrow c) \Rightarrow ((a \wedge b) \Rightarrow c) \approx \top$  **nitpick** *<proof>*

Similarly, 'weakening the consequent' is valid only under certain conditions.

**lemma** *WC'*:  $Fr-1b \mathcal{F} \implies \forall a b c. a \Rightarrow c \preceq a \Rightarrow (c \vee b)$  *<proof>*

**lemma** *WC*:  $Fr-1b \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b c. (a \Rightarrow c) \Rightarrow (a \Rightarrow (c \vee b)) \approx \top$  *<proof>*

**lemma** *Fr-1a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall a b c. a \Rightarrow c \preceq a \Rightarrow (c \vee b)$  **nitpick** *<proof>*

**lemma** *Fr-1a*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies \forall a b c. (a \Rightarrow c) \Rightarrow (a \Rightarrow (c \vee b)) \approx \top$  **nitpick** *<proof>*

end

**theory** *ex-subminimal-logics*

**imports** *topo-negation-conditions topo-strict-implication*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 11 Example application: Subintuitionistic and subminimal logics

In this section we examine some special paracomplete logics. The idea is to illustrate an approach by means of which we can obtain logics which are boldly paracomplete and (non-boldly) paraconsistent at the same time, Johansson's 'minimal logic' [7] being the paradigmatic case we aim at modelling.

Drawing upon the literature on Johanson's minimal logic, we introduce an unconstrained propositional constant  $Q$ , which we then employ to define a 'rigid' frontier operation  $\mathcal{F}'$ .

**consts**  $Q::\sigma$

**abbreviation**  $\mathcal{F}' \equiv \lambda X. Q$

**abbreviation**  $\mathcal{I}' \equiv \mathcal{I}_F \mathcal{F}'$

**abbreviation**  $\mathcal{C}' \equiv \mathcal{C}_F \mathcal{F}'$

**abbreviation**  $\mathcal{B}' \equiv \mathcal{B}_F \mathcal{F}'$

As defined,  $\mathcal{F}'$  (and its corresponding closure operation) satisfies several semantic conditions.

**lemma** *Fr-1*  $\mathcal{F}' \wedge$  *Fr-2*  $\mathcal{F}' \wedge$  *Fr-4*  $\mathcal{F}'$  *<proof>*

**lemma** *Cl-1*  $\mathcal{C}' \wedge$  *Cl-2*  $\mathcal{C}' \wedge$  *Cl-4*  $\mathcal{C}'$  *<proof>*

However *Fr-3* is not valid. In fact, adding it by hand would collapse into classical logic (making all sets clopen).

**lemma** *Fr-3*  $\mathcal{F}'$  **nitpick** *<proof>*

**lemma** *Cl-3*  $\mathcal{C}'$  **nitpick** *<proof>*

**lemma** *Fr-3*  $\mathcal{F}' \implies \forall A. \mathcal{F}'(A) \approx \perp$  *<proof>*

In order to obtain a paracomplete logic not validating ECQ, we define negation as follows,

**abbreviation** *neg-IC*:: $\sigma \Rightarrow \sigma$  ( $\neg$ ) **where**  $\neg A \equiv \mathcal{C}'(\mathcal{I}(-A))$

and observe that some plausible semantic properties obtain:

**lemma** *Q-def1*:  $\forall A. Q \approx \neg A \wedge \neg(\neg A)$  *<proof>*

**lemma** *Q-def2*: *Fr-1b*  $\mathcal{F} \implies \forall A. Q \approx \neg(A \vee \neg A)$  *<proof>*

**lemma** *neg-Idef*:  $\forall A. \neg A \approx \mathcal{I}(-A) \vee Q$  *<proof>*

**lemma** *neg-Cdef*: *Fr-2*  $\mathcal{F} \implies \forall A. \neg A \approx \mathcal{C}(A) \rightarrow Q$  *<proof>*

The negation so defined validates some properties corresponding to a (rather weak) paracomplete logic:

**lemma**  $\mathfrak{F} \mathcal{F} \implies TND \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies TNDw \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies TNDm \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies ECQ \neg$  **nitpick** *<proof>*

**lemma** *ECQw*:  $ECQw \neg$  *<proof>*

**lemma** *ECQm*:  $ECQm \neg$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies LNC \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies DNI \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies DNE \neg$  **nitpick** *<proof>*

**lemma** *CoPw*: *Fr-1b*  $\mathcal{F} \implies CoPw \neg$  *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies CoP1 \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies CoP2 \neg$  **nitpick** *<proof>*

**lemma**  $\mathfrak{F} \mathcal{F} \implies CoP3 \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies XCoP \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies DM3 \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $DM4: Fr-1a \mathcal{F} \implies DM4 \neg \langle proof \rangle$   
**lemma**  $Nor: Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies nNor \neg \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies nDNor \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoPw(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP1(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP2(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP3(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies DS1(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies DS2(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

Moreover, we cannot have both DNI and DNE without validating ECQ (thus losing paraconsistency).

**lemma**  $DNI \neg \wedge DNE \neg \longrightarrow ECQ \neg \langle proof \rangle$

However, we can have all of De Morgan laws while keeping (non-bold) paraconsistency.

**lemma**  $\sim ECQ \neg \wedge DM1 \neg \wedge DM2 \neg \wedge DM3 \neg \wedge DM4 \neg \wedge \mathfrak{F} \mathcal{F} \mathbf{nitpick}[satisfy, card w=3] \langle proof \rangle$

Below we combine negation with strict implication and verify some interesting properties. For instance, the following are not valid (and cannot become valid by adding semantic restrictions).

**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (\neg a \Rightarrow (a \Rightarrow b)) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (\neg a \rightarrow (a \rightarrow b)) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \wedge \neg a \Rightarrow b) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \wedge \neg a \rightarrow b) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \Rightarrow (b \vee \neg b)) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \rightarrow (b \vee \neg b)) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. (a \Rightarrow \neg a) \Rightarrow \neg a \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. (a \rightarrow \neg a) \rightarrow \neg a \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \wedge \neg a) \Rightarrow b \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \wedge \neg a) \rightarrow b \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. a \Rightarrow (b \vee \neg b) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. a \rightarrow (b \vee \neg b) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \leftrightarrow b) \Rightarrow (\neg a \leftrightarrow \neg b) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \leftrightarrow b) \rightarrow (\neg a \leftrightarrow \neg b) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \Rightarrow b) \wedge (a \Rightarrow \neg b) \Rightarrow \neg a \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a b. (a \rightarrow b) \wedge (a \rightarrow \neg b) \Rightarrow \neg a \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. (\neg a \Rightarrow \perp) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. (\neg a \rightarrow \perp) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. (\neg a \Rightarrow \neg(\neg\top)) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. (\neg a \rightarrow \neg(\neg\top)) \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. \neg(\neg(\neg a)) \Rightarrow \neg a \approx \top \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies \forall a. \neg(\neg(\neg a)) \rightarrow \neg a \approx \top \mathbf{nitpick} \langle proof \rangle$

The (weak) local contraposition axiom is indeed valid under appropriate conditions.

**lemma**  $lCoPw: Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies lCoPw(\Rightarrow) \neg \langle proof \rangle$   
**lemma**  $lCoPw-strict: \mathfrak{F} \mathcal{F} \implies \forall a b. (a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a) \approx \top \langle proof \rangle$

However, other (local) contraposition axioms are not valid.

**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP1(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP2(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies lCoP3(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

And this time no variant of disjunctive syllogism is valid.

**lemma**  $\mathfrak{F} \mathcal{F} \implies DS1(\implies) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies DS2(\implies) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies DS2(\implies) \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma**  $\mathfrak{F} \mathcal{F} \implies DS4(\implies) \neg \mathbf{nitpick} \langle proof \rangle$

Interestingly, one of the local contraposition axioms (lCoP1) follows from DNI.

**lemma** *DNI-lCoP1*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies Fr-4 \mathcal{F} \implies DNI \neg \longrightarrow lCoP1(\implies) \neg \langle proof \rangle$

This entails some other interesting results.

**lemma** *DNI-CoP1*:  $Fr-1b \mathcal{F} \implies DNI \neg \implies CoP1 \neg \langle proof \rangle$   
**lemma** *CoP1-LNC*:  $CoP1 \neg \implies LNC \neg \langle proof \rangle$   
**lemma** *DNI-LNC*:  $Fr-1b \mathcal{F} \implies DNI \neg \implies LNC \neg \langle proof \rangle$

The following variants of modus tollens also obtain.

**lemma** *MT*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. (a \implies b) \wedge \neg b \preceq \neg a \langle proof \rangle$   
**lemma** *MT'*:  $Fr-1 \mathcal{F} \implies Fr-2 \mathcal{F} \implies Fr-3 \mathcal{F} \implies \forall a b. ((a \implies b) \wedge \neg b) \Rightarrow \neg a \approx \top \langle proof \rangle$

We now semantically characterize (an approximation of) Johansson's Minimal Logic along with some exemplary 'subminimal' logics (observing that many more are possible). We check some relevant properties.

**abbreviation** *JML*  $\equiv \mathfrak{F} \mathcal{F} \wedge DNI \neg$   
**abbreviation** *SML1*  $\equiv \mathfrak{F} \mathcal{F}$   
**abbreviation** *SML2*  $\equiv Fr-1 \mathcal{F} \wedge Fr-2 \mathcal{F} \wedge Fr-3 \mathcal{F}$   
**abbreviation** *SML3*  $\equiv Fr-1 \mathcal{F}$   
**abbreviation** *SML4*  $\equiv Fr-1b \mathcal{F}$

TND:

**lemma** *JML*  $\implies TND \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma** *JML*  $\implies TNDw \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma** *JML*  $\implies TNDm \neg \mathbf{nitpick} \langle proof \rangle$

ECQ:

**lemma** *JML*  $\implies ECQ \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma** *ECQw*  $\neg \langle proof \rangle$   
**lemma** *ECQm*  $\neg \langle proof \rangle$

LNC:

**lemma** *JML*  $\implies LNC \neg \langle proof \rangle$   
**lemma** *SML1*  $\implies LNC \neg \mathbf{nitpick} \langle proof \rangle$

(r)DNI/DNE:

**lemma** *JML*  $\implies DNI \neg \langle proof \rangle$   
**lemma** *SML1*  $\implies rDNI \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma** *JML*  $\implies rDNE \neg \mathbf{nitpick} \langle proof \rangle$

CoP/MT:

**lemma** *SML4*  $\implies CoPw \neg \langle proof \rangle$   
**lemma** *JML*  $\implies CoP1 \neg \langle proof \rangle$   
**lemma** *SML1*  $\implies MT1 \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma** *JML*  $\implies MT2 \neg \mathbf{nitpick} \langle proof \rangle$   
**lemma** *JML*  $\implies MT3 \neg \mathbf{nitpick} \langle proof \rangle$

XCoP:



**lemma**  $JML \Rightarrow XCoP \neg \mathbf{nitpick} \langle proof \rangle$

DM3/4:

**lemma**  $JML \Rightarrow DM3 \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $SML3 \Rightarrow DM4 \neg \langle proof \rangle$

**lemma**  $SML4 \Rightarrow DM4 \neg \mathbf{nitpick} \langle proof \rangle$

nNor/nDNor:

**lemma**  $SML2 \Rightarrow nNor \neg \langle proof \rangle$

**lemma**  $SML3 \Rightarrow nNor \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow nDNor \neg \mathbf{nitpick} \langle proof \rangle$

lCoP classical:

**lemma**  $JML \Rightarrow lCoPw(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lCoP1(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lCoP2(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lCoP3(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

DS classical:

**lemma**  $JML \Rightarrow DS1(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow DS2(\rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

lCoP strict:

**lemma**  $SML1 \Rightarrow lCoPw(\Rightarrow) \neg \langle proof \rangle$

**lemma**  $SML2 \Rightarrow lCoPw(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lCoP1(\Rightarrow) \neg \langle proof \rangle$

**lemma**  $SML1 \Rightarrow lCoP1(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lCoP2(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lCoP3(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

lMT strict:

**lemma**  $SML2 \Rightarrow lMT0(\Rightarrow) \neg \langle proof \rangle$

**lemma**  $SML3 \Rightarrow lMT0(\Rightarrow) \neg \langle proof \rangle$

**lemma**  $SML4 \Rightarrow lMT0(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lMT1(\Rightarrow) \neg \langle proof \rangle$

**lemma**  $SML1 \Rightarrow lMT1(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lMT2(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow lMT3(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

DS strict:

**lemma**  $JML \Rightarrow DS1(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow DS2(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow DS3(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**lemma**  $JML \Rightarrow DS4(\Rightarrow) \neg \mathbf{nitpick} \langle proof \rangle$

**end**

**theory** *topo-derivative-algebra*

**imports** *topo-operators-derivative*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 12 Derivative algebra

The closure of a set  $A$  ( $\mathcal{C}(A)$ ) can be seen as the set  $A$  augmented by (i) its boundary points, or (ii) its accumulation/limit points. We explore the second variant by drawing on the notion of derivative algebra.

Declares a primitive (unconstrained) derivative (aka. derived-set) operation and defines others from it.

**consts**  $\mathcal{D}::\sigma\Rightarrow\sigma$   
**abbreviation**  $\mathcal{I} \equiv \mathcal{I}_D \mathcal{D}$  — interior  
**abbreviation**  $\mathcal{C} \equiv \mathcal{C}_D \mathcal{D}$  — closure  
**abbreviation**  $\mathcal{B} \equiv \mathcal{B}_D \mathcal{D}$  — border  
**abbreviation**  $\mathcal{F} \equiv \mathcal{F}_D \mathcal{D}$  — frontier  
**abbreviation**  $\mathcal{K} \equiv \mathcal{K}_D \mathcal{D}$  — coherence

### 12.1 Basic properties

Verifies minimal conditions under which operators resulting from conversion functions coincide.

**lemma**  $ICdual: \mathcal{I} \equiv \mathcal{C}^d$   $\langle proof \rangle$   
**lemma**  $ICdual': \mathcal{C} \equiv \mathcal{I}^d$   $\langle proof \rangle$   
**lemma**  $BI-rel: \mathcal{B} \equiv \mathcal{B}_I \mathcal{I}$   $\langle proof \rangle$   
**lemma**  $IB-rel: \mathcal{I} \equiv \mathcal{I}_B \mathcal{B}$   $\langle proof \rangle$   
**lemma**  $BC-rel: \mathcal{B} \equiv \mathcal{B}_C \mathcal{C}$   $\langle proof \rangle$   
**lemma**  $CB-rel: \mathcal{C} \equiv \mathcal{C}_B \mathcal{B}$   $\langle proof \rangle$   
**lemma**  $FI-rel: \mathcal{F} \equiv \mathcal{F}_I \mathcal{I}$   $\langle proof \rangle$   
**lemma**  $FC-rel: \mathcal{F} \equiv \mathcal{F}_C \mathcal{C}$   $\langle proof \rangle$   
**lemma**  $FB-rel: \mathcal{F} \equiv \mathcal{F}_B \mathcal{B}$   $\langle proof \rangle$

Recall that derivative and coherence operations cannot be obtained from either interior, closure, border nor frontier. The derivative operation can indeed be seen as being more fundamental than the other ones.

Fixed-point and other operators are interestingly related.

**lemma**  $fp1: \mathcal{I}^{fp} \equiv \mathcal{B}^c$   $\langle proof \rangle$   
**lemma**  $fp2: \mathcal{B}^{fp} \equiv \mathcal{I}^c$   $\langle proof \rangle$   
**lemma**  $fp3: \mathcal{C}^{fp} \equiv \mathcal{B}^d$   $\langle proof \rangle$   
**lemma**  $fp4: (\mathcal{B}^d)^{fp} \equiv \mathcal{C}$   $\langle proof \rangle$   
**lemma**  $fp5: \mathcal{F}^{fp} \equiv \mathcal{B} \sqcup (\mathcal{C}^c)$   $\langle proof \rangle$   
**lemma**  $fp6: \mathcal{D}^{fp} \equiv \mathcal{K} \sqcup (\mathcal{C}^c)$   $\langle proof \rangle$

Different inter-relations (some redundant ones are kept to help the provers).

**lemma**  $monI: Der-1b \mathcal{D} \Longrightarrow MONO \mathcal{I}$   $\langle proof \rangle$   
**lemma**  $monC: Der-1b \mathcal{D} \Longrightarrow MONO \mathcal{C}$   $\langle proof \rangle$   
**lemma**  $pB1: \forall A. \mathcal{B} A \approx A \leftarrow \mathcal{I} A$   $\langle proof \rangle$   
**lemma**  $pB2: \forall A. \mathcal{B} A \approx A \wedge \mathcal{F} A$   $\langle proof \rangle$   
**lemma**  $pB3: \forall A. \mathcal{B}(-A) \approx -A \wedge \mathcal{F} A$   $\langle proof \rangle$   
**lemma**  $pB4: \forall A. \mathcal{B}(-A) \approx -A \wedge \mathcal{C} A$   $\langle proof \rangle$   
**lemma**  $pB5: Der-1b \mathcal{D} \Longrightarrow \forall A. \mathcal{B}(\mathcal{C} A) \preceq (\mathcal{B} A) \vee \mathcal{B}(-A)$   $\langle proof \rangle$   
**lemma**  $pF1: \forall A. \mathcal{F} A \approx \mathcal{C} A \leftarrow \mathcal{I} A$   $\langle proof \rangle$   
**lemma**  $pF2: \forall A. \mathcal{F} A \approx \mathcal{C} A \wedge \mathcal{C}(-A)$   $\langle proof \rangle$   
**lemma**  $pF3: \forall A. \mathcal{F} A \approx \mathcal{B} A \vee \mathcal{B}(-A)$   $\langle proof \rangle$   
**lemma**  $pF4: Der-1 \mathcal{D} \Longrightarrow Der-4e \mathcal{D} \Longrightarrow \forall A. \mathcal{F}(\mathcal{I} A) \preceq \mathcal{F} A$   $\langle proof \rangle$   
**lemma**  $pF5: Der-1 \mathcal{D} \Longrightarrow Der-4e \mathcal{D} \Longrightarrow \forall A. \mathcal{F}(\mathcal{C} A) \preceq \mathcal{F} A$   $\langle proof \rangle$   
**lemma**  $pA1: \forall A. A \approx \mathcal{I} A \vee \mathcal{B} A$   $\langle proof \rangle$

**lemma**  $pA2$ :  $\forall A. A \approx C A \leftarrow B(-A)$   $\langle proof \rangle$

**lemma**  $pC1$ :  $\forall A. C A \approx A \vee B(-A)$   $\langle proof \rangle$

**lemma**  $pC2$ :  $\forall A. C A \approx A \vee F A$   $\langle proof \rangle$

**lemma**  $pI1$ :  $\forall A. I A \approx A \leftarrow B A$   $\langle proof \rangle$

**lemma**  $pI2$ :  $\forall A. I A \approx A \leftarrow F A$   $\langle proof \rangle$

**lemma**  $IC\text{-}imp$ :  $Der\text{-}1 \mathcal{D} \implies Der\text{-}3 \mathcal{D} \implies \forall A B. I(A \rightarrow B) \preceq C A \rightarrow C B$   $\langle proof \rangle$

Define some fixed-point predicates and prove some properties.

**abbreviation**  $openset$  ( $Op$ ) **where**  $Op A \equiv fp I A$

**abbreviation**  $closedset$  ( $Cl$ ) **where**  $Cl A \equiv fp C A$

**abbreviation**  $borderset$  ( $Br$ ) **where**  $Br A \equiv fp B A$

**abbreviation**  $frontierset$  ( $Fr$ ) **where**  $Fr A \equiv fp F A$

**lemma**  $Int\text{-}Open$ :  $Der\text{-}1a \mathcal{D} \implies Der\text{-}4e \mathcal{D} \implies \forall A. Op(I A)$   $\langle proof \rangle$

**lemma**  $Cl\text{-}Closed$ :  $Der\text{-}1a \mathcal{D} \implies Der\text{-}4e \mathcal{D} \implies \forall A. Cl(C A)$   $\langle proof \rangle$

**lemma**  $Br\text{-}Border$ :  $Der\text{-}1b \mathcal{D} \implies \forall A. Br(B A)$   $\langle proof \rangle$

In contrast, there is no analogous fixed-point result for frontier:

**lemma**  $\mathfrak{D} \mathcal{D} \implies \forall A. Fr(F A)$  **nitpick**  $\langle proof \rangle$

**lemma**  $OpCl\text{dual}$ :  $\forall A. Cl A \longleftrightarrow Op(-A)$   $\langle proof \rangle$

**lemma**  $ClOp\text{dual}$ :  $\forall A. Op A \longleftrightarrow Cl(-A)$   $\langle proof \rangle$

**lemma**  $Fr\text{-}ClBr$ :  $\forall A. Fr(A) = (Cl(A) \wedge Br(A))$   $\langle proof \rangle$

**lemma**  $Cl\text{-}F$ :  $Der\text{-}1 \mathcal{D} \implies Der\text{-}4e \mathcal{D} \implies \forall A. Cl(F A)$   $\langle proof \rangle$

## 12.2 Further properties

The definitions and theorems below are well known in the literature (e.g. [9]). Here we uncover the minimal conditions under which they hold (taking derivative operation as primitive).

**lemma**  $Cl\text{-}Bzero$ :  $\forall A. Cl A \longleftrightarrow B(-A) \approx \perp$   $\langle proof \rangle$

**lemma**  $Op\text{-}Bzero$ :  $\forall A. Op A \longleftrightarrow B A \approx \perp$   $\langle proof \rangle$

**lemma**  $Br\text{-}boundary$ :  $\forall A. Br(A) \longleftrightarrow I A \approx \perp$   $\langle proof \rangle$

**lemma**  $Fr\text{-}nowhereDense$ :  $\forall A. Fr(A) \longrightarrow I(C A) \approx \perp$   $\langle proof \rangle$

**lemma**  $Cl\text{-}FB$ :  $\forall A. Cl A \longleftrightarrow F A \approx B A$   $\langle proof \rangle$

**lemma**  $Op\text{-}FB$ :  $\forall A. Op A \longleftrightarrow F A \approx B(-A)$   $\langle proof \rangle$

**lemma**  $Clopen\text{-}Fzero$ :  $\forall A. Cl A \wedge Op A \longleftrightarrow F A \approx \perp$   $\langle proof \rangle$

**lemma**  $Int\text{-}sup\text{-}closed$ :  $Der\text{-}1b \mathcal{D} \implies supremum\text{-}closed (\lambda A. Op A)$   $\langle proof \rangle$

**lemma**  $Int\text{-}meet\text{-}closed$ :  $Der\text{-}1a \mathcal{D} \implies meet\text{-}closed (\lambda A. Op A)$   $\langle proof \rangle$

**lemma**  $Int\text{-}inf\text{-}closed$ :  $Der\text{-}inf \mathcal{D} \implies infimum\text{-}closed (\lambda A. Op A)$   $\langle proof \rangle$

**lemma**  $Cl\text{-}inf\text{-}closed$ :  $Der\text{-}1b \mathcal{D} \implies infimum\text{-}closed (\lambda A. Cl A)$   $\langle proof \rangle$

**lemma**  $Cl\text{-}join\text{-}closed$ :  $Der\text{-}1a \mathcal{D} \implies join\text{-}closed (\lambda A. Cl A)$   $\langle proof \rangle$

**lemma**  $Cl\text{-}sup\text{-}closed$ :  $Der\text{-}inf \mathcal{D} \implies supremum\text{-}closed (\lambda A. Cl A)$   $\langle proof \rangle$

**lemma**  $Br\text{-}inf\text{-}closed$ :  $Der\text{-}1b \mathcal{D} \implies infimum\text{-}closed (\lambda A. Br A)$   $\langle proof \rangle$

**lemma**  $Fr\text{-}inf\text{-}closed$ :  $Der\text{-}1b \mathcal{D} \implies infimum\text{-}closed (\lambda A. Fr A)$   $\langle proof \rangle$

**lemma**  $Br\text{-}Fr\text{-}join$ :  $Der\text{-}1 \mathcal{D} \implies Der\text{-}4e \mathcal{D} \implies \forall A B. Br A \wedge Fr B \longrightarrow Br(A \vee B)$   $\langle proof \rangle$

**lemma**  $Fr\text{-}join\text{-}closed$ :  $Der\text{-}1 \mathcal{D} \implies Der\text{-}4e \mathcal{D} \implies join\text{-}closed (\lambda A. Fr A)$   $\langle proof \rangle$

Introduces a predicate for indicating that two sets are disjoint and proves some properties.

**abbreviation**  $Disj A B \equiv A \wedge B \approx \perp$

**lemma**  $Disj\text{-}comm$ :  $\forall A B. Disj A B \longrightarrow Disj B A$   $\langle proof \rangle$

**lemma**  $Disj\text{-}IF$ :  $\forall A. Disj (I A) (F A)$   $\langle proof \rangle$

**lemma** *Disj-B*:  $\forall A. \text{Disj } (\mathcal{B} A) (\mathcal{B}(-A)) \langle \text{proof} \rangle$   
**lemma** *Disj-I*:  $\forall A. \text{Disj } (\mathcal{I} A) (-A) \langle \text{proof} \rangle$   
**lemma** *Disj-BCI*:  $\forall A. \text{Disj } (\mathcal{B}(\mathcal{C} A)) (\mathcal{I}(-A)) \langle \text{proof} \rangle$   
**lemma** *Disj-CBI*:  $\text{Der-1b } \mathcal{D} \implies \text{Der-4e } \mathcal{D} \implies \forall A. \text{Disj } (\mathcal{C}(\mathcal{B}(-A))) (\mathcal{I}(-A)) \langle \text{proof} \rangle$

Introduce a predicate for indicating that two sets are separated and proves some properties.

**definition** *Sep*  $A B \equiv \text{Disj } (\mathcal{C} A) B \wedge \text{Disj } (\mathcal{C} B) A$

**lemma** *Sep-comm*:  $\forall A B. \text{Sep } A B \longrightarrow \text{Sep } B A \langle \text{proof} \rangle$   
**lemma** *Sep-disj*:  $\forall A B. \text{Sep } A B \longrightarrow \text{Disj } A B \langle \text{proof} \rangle$   
**lemma** *Sep-I*:  $\text{Der-1 } \mathcal{D} \implies \text{Der-4e } \mathcal{D} \implies \forall A. \text{Sep } (\mathcal{I} A) (\mathcal{I} (-A)) \langle \text{proof} \rangle$

**lemma** *Sep-sub*:  $\text{Der-1b } \mathcal{D} \implies \forall A B C D. \text{Sep } A B \wedge C \preceq A \wedge D \preceq B \longrightarrow \text{Sep } C D \langle \text{proof} \rangle$   
**lemma** *Sep-Cl-diff*:  $\text{Der-1b } \mathcal{D} \implies \forall A B. \text{Cl}(A) \wedge \text{Cl}(B) \longrightarrow \text{Sep } (A \leftarrow B) (B \leftarrow A) \langle \text{proof} \rangle$   
**lemma** *Sep-Op-diff*:  $\text{Der-1b } \mathcal{D} \implies \forall A B. \text{Op}(A) \wedge \text{Op}(B) \longrightarrow \text{Sep } (A \leftarrow B) (B \leftarrow A) \langle \text{proof} \rangle$   
**lemma** *Sep-Cl*:  $\forall A B. \text{Cl}(A) \wedge \text{Cl}(B) \wedge \text{Disj } A B \longrightarrow \text{Sep } A B \langle \text{proof} \rangle$   
**lemma** *Sep-Op*:  $\text{Der-1b } \mathcal{D} \implies \forall A B. \text{Op}(A) \wedge \text{Op}(B) \wedge \text{Disj } A B \longrightarrow \text{Sep } A B \langle \text{proof} \rangle$   
**lemma** *Der-1a*:  $\text{Der-1b } \mathcal{D} \implies \forall A B C. \text{Sep } A B \wedge \text{Sep } A C \longrightarrow \text{Sep } A (B \vee C) \langle \text{proof} \rangle$

Verifies a neighborhood-based definition of interior.

**definition** *nbhd*  $A p \equiv \exists E. E \preceq A \wedge \text{Op}(E) \wedge (E p)$   
**lemma** *nbhd-def2*:  $\text{Der-1 } \mathcal{D} \implies \text{Der-4e } \mathcal{D} \implies \forall A p. (\text{nbhd } A p) = (\mathcal{I} A p) \langle \text{proof} \rangle$

**lemma** *I-def'-lr'*:  $\forall A p. (\mathcal{I} A p) \longrightarrow (\exists E. (\mathcal{I} E p) \wedge E \preceq A) \langle \text{proof} \rangle$   
**lemma** *I-def'-rl'*:  $\text{Der-1b } \mathcal{D} \implies \forall A p. (\mathcal{I} A p) \longleftarrow (\exists E. (\mathcal{I} E p) \wedge E \preceq A) \langle \text{proof} \rangle$   
**lemma** *I-def'*:  $\text{Der-1b } \mathcal{D} \implies \forall A p. (\mathcal{I} A p) \longleftrightarrow (\exists E. (\mathcal{I} E p) \wedge E \preceq A) \langle \text{proof} \rangle$

Explore the Barcan and converse Barcan formulas.

**lemma** *Barcan-I*:  $\text{Der-inf } \mathcal{D} \implies \forall P. (\forall x. \mathcal{I}(P x)) \preceq \mathcal{I}(\forall x. P x) \langle \text{proof} \rangle$   
**lemma** *Barcan-C*:  $\text{Der-inf } \mathcal{D} \implies \forall P. \mathcal{C}(\exists x. P x) \preceq (\exists x. \mathcal{C}(P x)) \langle \text{proof} \rangle$   
**lemma** *CBarcan-I*:  $\text{Der-1b } \mathcal{D} \implies \forall P. \mathcal{I}(\forall x. P x) \preceq (\forall x. \mathcal{I}(P x)) \langle \text{proof} \rangle$   
**lemma** *CBarcan-C*:  $\text{Der-1b } \mathcal{D} \implies \forall P. (\exists x. \mathcal{C}(P x)) \preceq \mathcal{C}(\exists x. P x) \langle \text{proof} \rangle$

**end**

**theory** *ex-LFUs*

**imports** *topo-derivative-algebra sse-operation-negative*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 13 Example application: Logics of Formal Undeterminedness (LFUs)

The LFUs [10] [4] are a family of paracomplete logics featuring a 'determinedness' operator  $\circ$  that can be used to recover some classical properties of negation (in particular TND). LFUs behave in a sense dually to LFIs. Both can be semantically embedded as extensions of Boolean algebras. Here we show how to semantically embed LFUs as derivative algebras.

(We rename (classical) meta-logical negation to avoid terminological confusion)

**abbreviation** *cneg::bool $\Rightarrow$ bool* ( $\sim$  [40]40) **where**  $\sim\varphi \equiv \neg\varphi$

Logical validity can be defined as truth in all worlds/points (i.e. as denoting the top element)

**abbreviation** *gtrue:: $\sigma\Rightarrow$ bool* ( $\Vdash$  -) **where**  $\Vdash A \equiv \forall w. A w$

**lemma** *gtrue-def*:  $\vdash A \equiv A \approx \top$  *<proof>*

As for LFIs, we focus on the local (truth-degree preserving) notion of logical consequence.

**abbreviation** *conseq-local1*:: $\sigma \Rightarrow \sigma \Rightarrow \text{bool}$  ( $[- \vdash -]$ ) **where**  $[A \vdash B] \equiv A \preceq B$

**abbreviation** *conseq-local2*:: $\sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow \text{bool}$  ( $[-, - \vdash -]$ ) **where**  $[A_1, A_2 \vdash B] \equiv A_1 \wedge A_2 \preceq B$

**abbreviation** *conseq-local12*:: $\sigma \Rightarrow \sigma \Rightarrow \sigma \Rightarrow \text{bool}$  ( $[- \vdash -, -]$ ) **where**  $[A \vdash B_1, B_2] \equiv A \preceq B_1 \vee B_2$

For LFUs we use the interior-based negation previously defined (taking derivative as primitive).

**definition** *ineg*:: $\sigma \Rightarrow \sigma$  ( $\neg$ ) **where**  $\neg A \equiv \mathcal{I}(-A)$

**declare** *ineg-def*[*conn*]

In terms of the derivative operator the negation looks as follows:

**lemma** *ineg-prop*:  $\neg A \approx -(\mathcal{D} A) \leftarrow A$  *<proof>*

This negation is of course paracomplete.

**lemma**  $\vdash a \vee \neg a$  **nitpick** *<proof>*

We define two pairs of in/determinedness operators and show how they relate to each other.

**abbreviation** *op-det*:: $\sigma \Rightarrow \sigma$  ( $\circ-$  [57]58) **where**  $\circ A \equiv \mathcal{B}^d A$

**abbreviation** *op-ind*:: $\sigma \Rightarrow \sigma$  ( $\cdot-$  [57]58) **where**  $\cdot A \equiv -\circ A$

**lemma** *op-det-def*:  $\circ a \approx a \vee \neg a$  *<proof>*

**lemma** *Prop1*:  $\circ A \approx \mathcal{C}^{fp} A$  *<proof>*

**lemma** *Prop2*:  $Op A \longleftrightarrow \circ - A \approx \top$  *<proof>*

**lemma** *Prop3*:  $Op A \longleftrightarrow \cdot - A \approx \perp$  *<proof>*

**lemma** *Prop4*:  $Cl A \longleftrightarrow \circ A \approx \top$  *<proof>*

**lemma** *Prop5*:  $Cl A \longleftrightarrow \cdot A \approx \perp$  *<proof>*

Analogously as for LFIs, LFUs provide means for recovering the principle of excluded middle.

**lemma**  $[\Gamma \vdash \cdot a, a \vee \neg a]$  *<proof>*

**lemma**  $[\Gamma, \circ a \vdash a \vee \neg a]$  *<proof>*

**lemma** *TNDm*( $\neg$ ) **nitpick** *<proof>*

**lemma** *ECQ*( $\neg$ ) *<proof>*

**lemma** *Der-3*  $\mathcal{D} \Rightarrow LNC(\neg)$  *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow DNI(\neg)$  **nitpick** *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow DNE(\neg)$  **nitpick** *<proof>*

**lemma** *Der-1b*  $\mathcal{D} \Rightarrow CoPw(\neg)$  *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow CoP1(\neg)$  **nitpick** *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow CoP2(\neg)$  **nitpick** *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow CoP3(\neg)$  **nitpick** *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow DM3(\neg)$  **nitpick** *<proof>*

**lemma** *Der-1a*  $\mathcal{D} \Rightarrow DM4(\neg)$  *<proof>*

**lemma** *Der-3*  $\mathcal{D} \Rightarrow nNor(\neg)$  *<proof>*

**lemma** *nDNor*( $\neg$ ) *<proof>*

**lemma** *Der-1b*  $\mathcal{D} \Rightarrow MT0(\neg)$  *<proof>*

**lemma** *Der-1b*  $\mathcal{D} \Rightarrow \text{Der-3 } \mathcal{D} \Rightarrow MT1(\neg)$  *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow MT2(\neg)$  **nitpick** *<proof>*

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow MT3(\neg)$  **nitpick** *<proof>*

We show how all local contraposition variants (lCoP) can be recovered using the determinedness operator. Observe that we can recover in the same way other (weaker) properties: CoP, MT and DNI/DNE (local & global).

**lemma**  $\mathfrak{D} \mathcal{D} \Rightarrow lCoPw(\rightarrow)(\neg)$  **nitpick** *<proof>*

**lemma** *det-lcop1*:  $[\circ a, a \rightarrow b \vdash \neg b \rightarrow \neg a]$  *<proof>*  
**lemma**  $\mathcal{D} \mathcal{D} \implies lCoP1(\rightarrow)(\neg)$  **nitpick** *<proof>*  
**lemma** *det-lcop2*:  $[\circ a, a \rightarrow \neg b \vdash b \rightarrow \neg a]$  *<proof>*  
**lemma**  $\mathcal{D} \mathcal{D} \implies lCoP2(\rightarrow)(\neg)$  **nitpick** *<proof>*  
**lemma** *det-lcop3*:  $[\circ a, \neg a \rightarrow b \vdash \neg b \rightarrow a]$  *<proof>*  
**lemma**  $\mathcal{D} \mathcal{D} \implies lCoP3(\rightarrow)(\neg)$  **nitpick** *<proof>*  
**lemma** *det-lcop4*:  $[\circ a, \neg a \rightarrow \neg b \vdash b \rightarrow a]$  *<proof>*

Disjunctive syllogism (DS).

**lemma** *DS1*( $\rightarrow$ )( $\neg$ ) *<proof>*  
**lemma**  $\mathcal{D} \mathcal{D} \implies DS2(\rightarrow)(\neg)$  **nitpick** *<proof>*  
**lemma** *det-ds2*:  $[\circ a, \neg a \rightarrow b \vdash a \vee b]$  *<proof>*

**end**

**theory** *topo-border-algebra*

**imports** *topo-operators-basic*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 14 Border algebra

We define a border algebra in an analogous fashion to the well-known closure/interior algebras. We also verify a few interesting properties.

Declares a primitive (unconstrained) border operation and defines others from it.

**consts**  $\mathcal{B}::\sigma \Rightarrow \sigma$   
**abbreviation**  $\mathcal{I} \equiv \mathcal{I}_B \mathcal{B}$  — interior  
**abbreviation**  $\mathcal{C} \equiv \mathcal{C}_B \mathcal{B}$  — closure  
**abbreviation**  $\mathcal{F} \equiv \mathcal{F}_B \mathcal{B}$  — frontier

### 14.1 Basic properties

Verifies minimal conditions under which operators resulting from conversion functions coincide.

**lemma** *ICdual*:  $\mathcal{I} \equiv \mathcal{C}^d$  *<proof>*  
**lemma** *ICdual'*:  $\mathcal{C} \equiv \mathcal{I}^d$  *<proof>*  
**lemma** *FI-rel*:  $Br-1 \mathcal{B} \implies \mathcal{F} \equiv \mathcal{F}_I \mathcal{I}$  *<proof>*  
**lemma** *IF-rel*:  $Br-1 \mathcal{B} \implies \mathcal{I} \equiv \mathcal{I}_F \mathcal{F}$  *<proof>*  
**lemma** *FC-rel*:  $Br-1 \mathcal{B} \implies \mathcal{F} \equiv \mathcal{F}_C \mathcal{C}$  *<proof>*  
**lemma** *CF-rel*:  $Br-1 \mathcal{B} \implies \mathcal{C} \equiv \mathcal{C}_F \mathcal{F}$  *<proof>*  
**lemma** *BI-rel*:  $Br-1 \mathcal{B} \implies \mathcal{B} \equiv \mathcal{B}_I \mathcal{I}$  *<proof>*  
**lemma** *BC-rel*:  $Br-1 \mathcal{B} \implies \mathcal{B} \equiv \mathcal{B}_C \mathcal{C}$  *<proof>*  
**lemma** *BF-rel*:  $Br-1 \mathcal{B} \implies \mathcal{B} \equiv \mathcal{B}_F \mathcal{F}$  *<proof>*

Fixed-point and other operators are interestingly related.

**lemma** *fp1*:  $Br-1 \mathcal{B} \implies \mathcal{I}^{fp} \equiv \mathcal{B}^c$  *<proof>*  
**lemma** *fp2*:  $Br-1 \mathcal{B} \implies \mathcal{B}^{fp} \equiv \mathcal{I}^c$  *<proof>*  
**lemma** *fp3*:  $Br-1 \mathcal{B} \implies \mathcal{C}^{fp} \equiv \mathcal{B}^d$  *<proof>*  
**lemma** *fp4*:  $Br-1 \mathcal{B} \implies (\mathcal{B}^d)^{fp} \equiv \mathcal{C}$  *<proof>*  
**lemma** *fp5*:  $Br-1 \mathcal{B} \implies \mathcal{F}^{fp} \equiv \mathcal{B} \sqcup (\mathcal{C}^c)$  *<proof>*

Define some fixed-point predicates and prove some properties.

**abbreviation** *openset* (*Op*) **where**  $Op A \equiv fp \mathcal{I} A$   
**abbreviation** *closedset* (*Cl*) **where**  $Cl A \equiv fp \mathcal{C} A$

**abbreviation** *borderset* (*Br*) **where**  $Br\ A \equiv fp\ \mathcal{B}\ A$   
**abbreviation** *frontierset* (*Fr*) **where**  $Fr\ A \equiv fp\ \mathcal{F}\ A$

**lemma** *Int-Open*:  $Br-1\ \mathcal{B} \implies Br-3\ \mathcal{B} \implies \forall A. Op(\mathcal{I}\ A) \langle proof \rangle$   
**lemma** *Cl-Closed*:  $Br-1\ \mathcal{B} \implies Br-3\ \mathcal{B} \implies \forall A. Cl(\mathcal{C}\ A) \langle proof \rangle$   
**lemma** *Br-Border*:  $Br-1\ \mathcal{B} \implies \forall A. Br(\mathcal{B}\ A) \langle proof \rangle$

In contrast, there is no analogous fixed-point result for frontier:

**lemma**  $\mathfrak{B}\ \mathcal{B} \implies \forall A. Fr(\mathcal{F}\ A)$  **nitpick**  $\langle proof \rangle$

**lemma** *OpClDual*:  $\forall A. Cl\ A \longleftrightarrow Op(-A) \langle proof \rangle$   
**lemma** *ClOpDual*:  $\forall A. Op\ A \longleftrightarrow Cl(-A) \langle proof \rangle$   
**lemma** *Fr-ClBr*:  $Br-1\ \mathcal{B} \implies \forall A. Fr(A) = (Cl(A) \wedge Br(A)) \langle proof \rangle$   
**lemma** *Cl-F*:  $Br-1\ \mathcal{B} \implies Br-3\ \mathcal{B} \implies \forall A. Cl(\mathcal{F}\ A) \langle proof \rangle$

**end**

**theory** *topo-closure-algebra*

**imports** *topo-operators-basic*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 15 Closure algebra

We define a topological Boolean algebra with a primitive closure operator and verify a few properties.

Declares a primitive (unconstrained) closure operation and defines others from it.

**consts**  $C::\sigma \Rightarrow \sigma$   
**abbreviation**  $\mathcal{I} \equiv C^d$  — interior  
**abbreviation**  $\mathcal{B} \equiv \mathcal{B}_C\ C$  — border  
**abbreviation**  $\mathcal{F} \equiv \mathcal{F}_C\ C$  — frontier

### 15.1 Basic properties

Verifies minimal conditions under which operators resulting from conversion functions coincide.

**lemma** *ICdual'*:  $C \equiv \mathcal{I}^d \langle proof \rangle$   
**lemma** *IB-rel*:  $Cl-2\ C \implies \mathcal{I} \equiv \mathcal{I}_B\ \mathcal{B} \langle proof \rangle$   
**lemma** *IF-rel*:  $Cl-2\ C \implies \mathcal{I} \equiv \mathcal{I}_F\ \mathcal{F} \langle proof \rangle$   
**lemma** *CB-rel*:  $Cl-2\ C \implies C \equiv \mathcal{C}_B\ \mathcal{B} \langle proof \rangle$   
**lemma** *CF-rel*:  $Cl-2\ C \implies C \equiv \mathcal{C}_F\ \mathcal{F} \langle proof \rangle$   
**lemma** *BI-rel*:  $\mathcal{B} \equiv \mathcal{B}_I\ \mathcal{I} \langle proof \rangle$   
**lemma** *BF-rel*:  $Cl-2\ C \implies \mathcal{B} \equiv \mathcal{B}_F\ \mathcal{F} \langle proof \rangle$   
**lemma** *FI-rel*:  $\mathcal{F} \equiv \mathcal{F}_I\ \mathcal{I} \langle proof \rangle$   
**lemma** *FB-rel*:  $Cl-2\ C \implies \mathcal{F} \equiv \mathcal{F}_B\ \mathcal{B} \langle proof \rangle$

Fixed-point and other operators are interestingly related.

**lemma** *fp1*:  $Cl-2\ C \implies \mathcal{I}^{fp} \equiv \mathcal{B}^c \langle proof \rangle$   
**lemma** *fp2*:  $Cl-2\ C \implies \mathcal{B}^{fp} \equiv \mathcal{I}^c \langle proof \rangle$   
**lemma** *fp3*:  $Cl-2\ C \implies \mathcal{C}^{fp} \equiv \mathcal{B}^d \langle proof \rangle$   
**lemma** *fp4*:  $Cl-2\ C \implies (\mathcal{B}^d)^{fp} \equiv C \langle proof \rangle$   
**lemma** *fp5*:  $Cl-2\ C \implies \mathcal{F}^{fp} \equiv \mathcal{B} \sqcup (C^c) \langle proof \rangle$

Define some fixed-point predicates and prove some properties.

**abbreviation** *openset* (*Op*) **where**  $Op\ A \equiv fp\ \mathcal{I}\ A$   
**abbreviation** *closedset* (*Cl*) **where**  $Cl\ A \equiv fp\ \mathcal{C}\ A$   
**abbreviation** *borderset* (*Br*) **where**  $Br\ A \equiv fp\ \mathcal{B}\ A$   
**abbreviation** *frontierset* (*Fr*) **where**  $Fr\ A \equiv fp\ \mathcal{F}\ A$

**lemma** *Int-Open*:  $Cl\text{-}4\ \mathcal{C} \implies \forall A. Op(\mathcal{I}\ A) \langle proof \rangle$   
**lemma** *Cl-Closed*:  $Cl\text{-}4\ \mathcal{C} \implies \forall A. Cl(\mathcal{C}\ A) \langle proof \rangle$   
**lemma** *Br-Border*:  $Cl\text{-}1b\ \mathcal{C} \implies \forall A. Br(\mathcal{B}\ A) \langle proof \rangle$

In contrast, there is no analogous fixed-point result for frontier:

**lemma**  $\mathcal{C}\ \mathcal{C} \implies \forall A. Fr(\mathcal{F}\ A)$  **nitpick**  $\langle proof \rangle$

**lemma** *OpClDual*:  $\forall A. Cl\ A \longleftrightarrow Op(-A) \langle proof \rangle$   
**lemma** *ClOpDual*:  $\forall A. Op\ A \longleftrightarrow Cl(-A) \langle proof \rangle$   
**lemma** *Fr-ClBr*:  $Cl\text{-}2\ \mathcal{C} \implies \forall A. Fr(A) = (Cl(A) \wedge Br(A)) \langle proof \rangle$   
**lemma** *Cl-F*:  $Cl\text{-}1b\ \mathcal{C} \implies Cl\text{-}2\ \mathcal{C} \implies Cl\text{-}4\ \mathcal{C} \implies \forall A. Cl(\mathcal{F}\ A) \langle proof \rangle$

**end**

**theory** *topo-interior-algebra*

**imports** *topo-operators-basic*

**begin**

**nitpick-params**[*assms=true, user-axioms=true, show-all, expect=genuine, format=3*]

## 16 Interior algebra

We define a topological Boolean algebra taking the interior operator as primitive and verify some properties.

Declares a primitive (unconstrained) interior operation and defines others from it.

**consts**  $\mathcal{I}::\sigma \Rightarrow \sigma$   
**abbreviation**  $\mathcal{C} \equiv \mathcal{I}^d$  — closure  
**abbreviation**  $\mathcal{B} \equiv \mathcal{B}_I\ \mathcal{I}$  — border  
**abbreviation**  $\mathcal{F} \equiv \mathcal{F}_I\ \mathcal{I}$  — frontier

### 16.1 Basic properties

Verifies minimal conditions under which operators resulting from conversion functions coincide.

**lemma** *ICdual*:  $\mathcal{I} \equiv \mathcal{C}^d \langle proof \rangle$   
**lemma** *IB-rel*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{I} \equiv \mathcal{I}_B\ \mathcal{B} \langle proof \rangle$   
**lemma** *IF-rel*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{I} \equiv \mathcal{I}_F\ \mathcal{F} \langle proof \rangle$   
**lemma** *CB-rel*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{C} \equiv \mathcal{C}_B\ \mathcal{B} \langle proof \rangle$   
**lemma** *CF-rel*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{C} \equiv \mathcal{C}_F\ \mathcal{F} \langle proof \rangle$   
**lemma** *BC-rel*:  $\mathcal{B} \equiv \mathcal{B}_C\ \mathcal{C} \langle proof \rangle$   
**lemma** *BF-rel*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{B} \equiv \mathcal{B}_F\ \mathcal{F} \langle proof \rangle$   
**lemma** *FC-rel*:  $\mathcal{F} \equiv \mathcal{F}_C\ \mathcal{C} \langle proof \rangle$   
**lemma** *FB-rel*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{F} \equiv \mathcal{F}_B\ \mathcal{B} \langle proof \rangle$

Fixed-point and other operators are interestingly related.

**lemma** *fp1*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{I}^{fp} \equiv \mathcal{B}^c \langle proof \rangle$   
**lemma** *fp2*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{B}^{fp} \equiv \mathcal{I}^c \langle proof \rangle$   
**lemma** *fp3*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{C}^{fp} \equiv \mathcal{B}^d \langle proof \rangle$   
**lemma** *fp4*:  $Int\text{-}2\ \mathcal{I} \implies (\mathcal{B}^d)^{fp} \equiv \mathcal{C} \langle proof \rangle$   
**lemma** *fp5*:  $Int\text{-}2\ \mathcal{I} \implies \mathcal{F}^{fp} \equiv \mathcal{B} \sqcup (\mathcal{C}^c) \langle proof \rangle$



Define some fixed-point predicates and prove some properties.

**abbreviation** *openset* ( $Op$ ) **where**  $Op\ A \equiv fp\ \mathcal{I}\ A$

**abbreviation** *closedset* ( $Cl$ ) **where**  $Cl\ A \equiv fp\ \mathcal{C}\ A$

**abbreviation** *borderset* ( $Br$ ) **where**  $Br\ A \equiv fp\ \mathcal{B}\ A$

**abbreviation** *frontierset* ( $Fr$ ) **where**  $Fr\ A \equiv fp\ \mathcal{F}\ A$

**lemma** *Int-Open*:  $Int\text{-}4\ \mathcal{I} \implies \forall A. Op(\mathcal{I}\ A) \langle proof \rangle$

**lemma** *Cl-Closed*:  $Int\text{-}4\ \mathcal{I} \implies \forall A. Cl(\mathcal{C}\ A) \langle proof \rangle$

**lemma** *Br-Border*:  $Int\text{-}1a\ \mathcal{I} \implies \forall A. Br(\mathcal{B}\ A) \langle proof \rangle$

In contrast, there is no analogous fixed-point result for frontier:

**lemma**  $\exists\ \mathcal{I} \implies \forall A. Fr(\mathcal{F}\ A)$  **nitpick**  $\langle proof \rangle$

**lemma** *OpClDual*:  $\forall A. Cl\ A \longleftrightarrow Op(-A) \langle proof \rangle$

**lemma** *ClOpDual*:  $\forall A. Op\ A \longleftrightarrow Cl(-A) \langle proof \rangle$

**lemma** *Fr-ClBr*:  $Int\text{-}2\ \mathcal{I} \implies \forall A. Fr(A) = (Cl(A) \wedge Br(A)) \langle proof \rangle$

**lemma** *Cl-F*:  $Int\text{-}1a\ \mathcal{I} \implies Int\text{-}2\ \mathcal{I} \implies Int\text{-}4\ \mathcal{I} \implies \forall A. Cl(\mathcal{F}\ A) \langle proof \rangle$

end

## References

- [1] C. Benzmüller. Universal (meta-)logical reasoning: Recent successes. *Science of Computer Programming*, 172:48–62, 2019. Preprint: <http://doi.org/10.13140/RG.2.2.11039.61609/2>.
- [2] C. Benzmüller and L. Paulson. Quantified multimodal logics in simple type theory. *Logica Universalis (Special Issue on Multimodal Logics)*, 7(1):7–20, 2013. Preprint: <http://christoph-benzmueller.de/papers/J23.pdf>.
- [3] W. Carnielli, M. E. Coniglio, and D. Fuenmayor. Logics of formal inconsistency enriched with replacement: an algebraic and modal account. *arXiv*, math.LO(2003.09522), 2020.
- [4] W. Carnielli, M. E. Coniglio, and J. Marcos. Logics of formal inconsistency. In *Handbook of philosophical logic*, pages 1–93. Springer, 2007.
- [5] L. Esakia. Intuitionistic logic and modality via topology. *Annals of Pure and Applied Logic*, 127(1-3):155–170, 2004.
- [6] F. Hausdorff. *Grundzüge der Mengenlehre*, volume 7. von Veit, 1914.
- [7] I. Johansson. Der Minimalkalkül, ein reduzierter intuitionistischer Formalismus. *Compositio mathematica*, 4:119–136, 1937.
- [8] K. Kuratowski. Sur l’opération  $\bar{A}$  de l’analysis situs. *Fundamenta Mathematicae*, 3(1):182–199, 1922.
- [9] K. Kuratowski. *Topology: Volume I*, volume 1. Elsevier, 2014.
- [10] J. Marcos. Nearly every normal modal logic is paranormal. *Logique et Analyse*, 48(189/192):279–300, 2005.
- [11] J. C. C. McKinsey and A. Tarski. The algebra of topology. *Annals of mathematics*, pages 141–191, 1944.

- [12] M. Zarycki. Quelques notions fondamentales de l'analysis situs au point de vue de l'algèbre de la logique. *Fundamenta Mathematicae*, 9(1):3–15, 1927. English translation by Mark Bowron: <https://www.researchgate.net/scientific-contributions/Miron-Zarycki-2016157096>.
- [13] M. Zarycki. Allgemeine Eigenschaften der cantorschen Kohärenzen. *Transactions of the American Mathematical Society*, 30(3):498–506, 1928. English translation by Mark Bowron: <https://www.researchgate.net/scientific-contributions/Miron-Zarycki-2016157096>.
- [14] M. Zarycki. Some properties of the derived set operation in abstract spaces. *Nauk. Zap. Ser. Fiz.-Mat.*, 5:22–33, 1947. English translation by Mark Bowron: <https://www.researchgate.net/scientific-contributions/Miron-Zarycki-2016157096>.