Journal of applied logics (Print) ISSN 2631-9810
Journal of applied logics (Online) ISSN 2631-9829

Journal of Applied Logics

The IfCoLog Journal of Logics and their Applications

Volume 12 • Issue 7 • December 2025

Available online at
www.collegepublications.co.uk/journals/ifcolog/
Free open access

JOURNAL OF APPLIED LOGICS - IFCOLOG
JOURNAL OF LOGICS AND THEIR APPLICATIONS

Volume 12, Number 7

December 2025

Disclaimer

Statements of fact and opinion in the articles in Journal of Applied Logics - IfCoLog Journal of Logics and their Applications (JALs-FLAP) are those of the respective authors and contributors and not of the JALs-FLAP. Neither College Publications nor the JALs-FLAP make any representation, express or implied, in respect of the accuracy of the material in this journal and cannot accept any legal responsibility or liability for any errors or omissions that may be made. The reader should make his/her own evaluation as to the appropriateness or otherwise of any experimental technique described.

ISBN 978-1-84890-500-9
ISSN (E) 2631-9829
ISSN (P) 2631-9810

College Publications
Scientific Director: Dov Gabbay
Managing Director: Jane Spurr

http://www.collegepublications.co.uk

Editorial Board

Scope and Submissions

This journal considers submission in all areas of pure and applied logic, including:

- pure logical systems
- proof theory
- constructive logic
- categorical logic
- modal and temporal logic
- model theory
- recursion theory
- type theory
- nominal theory
- nonclassical logics
- nonmonotonic logic
- numerical and uncertainty reasoning
- logic and AI
- foundations of logic programming
- belief change/revision
- systems of knowledge and belief
- logics and semantics of programming
- specification and verification
- agent theory
- databases
- dynamic logic
- quantum logic
- algebraic logic
- logic and cognition
- probabilistic logic
- logic and networks
- neuro-logical systems
- complexity
- argumentation theory
- logic and computation
- logic and language
- logic engineering
- knowledge-based systems
- automated reasoning
- knowledge representation
- logic in hardware and VLSI
- natural language
- concurrent computation
- planning

This journal will also consider papers on the application of logic in other subject areas: philosophy, cognitive science, physics etc. provided they have some formal content.

Submissions should be sent to Jane Spurr (jane@janespurr.net) as a pdf file, preferably compiled in LaTeX using the IFCoLog class file.

Contents

1811

1812

On Reductions of Hintikka Sets for Higher-Order Logic

Alexander Steen
Institute of Mathematics and Computer Science, University of Greifswald, Greifswald, Germany
alexander.steen@uni-greifswald.de

Christoph Benzmüller
Faculty Information Systems and Applied Computer Sciences, University of Bamberg, Bamberg, Germany; and Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany
christoph.benzmueller@uni-bamberg.de

Abstract

Steen's (2018) properties for Hintikka sets for Church's type theory based on primitive equality are reduced to the alternative, technically different, Hintikka set properties of Brown (2007). Using this reduction, a model existence theorem for Steen's properties is derived. In related work by Steen and Benzmüller (2021) this model existence result has been employed to prove completeness of the higher-order paramodulation calculus underlying the Leo-III prover.

1 Introduction

Abstract Consistency properties and Hintikka sets have been successfully used in the study of theoretical properties of proof calculi for classical higher order logic (HOL), also known as Church's type theory [12, 6]. In conjunction with associated model existence theorems, they have been used, for example, to establish Henkin-completeness results for various proof calculi for HOL [7, 11, 10, 9, 15]. Unfortunately, however, technically quite different definitions of abstract consistency

The work was partially supported by the German National Research Foundation (DFG) under grant BE 2501/11-1 (Leo-III). We thank the anonymous reviewer of this paper acting as a reviewer for the JSL, and prompting further revision). We thank an anonymous reviewer of [17] and [16] for very valuable feedback to this work (cf. Footnote 4). Also, we thank Geoff Sutcliffe for improving the quality of the manuscript.

have been developed and used in these works, due to a technical dependence on the assumed primitive logical connectives in the respective calculi. For example, the calculi and abstract consistency conditions studied in Benzmüller, Brown, and Kohlhase [7, 8] are based on negation, disjunction, and universal quantification, whereas Steen [15], in the tradition of Andrews [4], works with only primitive equality. Thus, despite their conceptual affinity, important semantic corollaries arising from syntax-related abstract consistency conditions and the corresponding Hintikka properties (e.g., model existence theorems) cannot be readily transferred between formalisms. This, in turn, needlessly complicates completeness proofs of (machine-oriented) calculi as they cannot simply interface with available results.[1]

Brown [11, 10] addresses this problem by introducing generalised abstract consistency properties where the primitive logical connectives can vary. In order to achieve this, however, Brown uses an extended syntactical formalism for HOL terms (introducing so-called *external* propositions that act as meta-terms) that adds another layer of complexity. In particular, it's not directly clear how to apply Brown's result to the usual formulations of HOL and its proof calculi. In this paper it is shown that Steen's properties can be reduced to Brown's. The theorem 5.1 in this paper paves the way for convenient reuse of results (e.g., model existence) from Brown's work [11, 10] in the context of Steen's setting. For this reduction to work, it is technically sufficient to establish the connection between the respective Hintikka properties as defined by Steen and Brown.

The model-existence result reported in this article was an essential prerequisite for proving the Henkin-completeness of the higher-order paramodulation calculus underlying the automated theorem proving system Leo-III; cf. the recent article by Steen and Benzmüller [17]. Up to the authors' knowledge, such a translation between different notions of Hintikka sets has not been discussed before in the literature. Moreover, it should be possible to adapt the reduction technique to obtain model existence results for various related, but technically different formalisms (including, e.g., correspondences for Andrews' V-complexes [1] for arbitrary signatures).

[1]The problem here is that for machine-oriented calculi there may be practically motivated design decisions (e.g., the choice of primitive logical connectives), that differ from the interfaces provided by relevant results from the literature. Without matching interfaces to these results the completeness proofs need to reiterate conceptually similar arguments in a slightly different setting, while the core argument essentially stays the same. The presented technique aims at removing one aspect of this unnecessary complex process, and to flexibly bridge to the (fixed) interfaces from literature to the specific needs at hand.

Paper structure.

In §2 some relevant notions of HOL are reviewed; this section is intended to keep this article sufficiently self-contained. In §3 the Hintikka properties as used by Brown are presented, and in §4 the related properties as used by Steen are shown, and various lemmas that are implied in this setting are proven. These lemmas then prepare the main reduction result of this work, Theorem 5.1, given in §5. A model existence theorem for Hintikka sets as defined by Steen is finally derived in §6.

2 Preliminaries

HOL is a logic based on the simply typed λ-calculus [13, 5]. Logical connectives (e.g., primitive equality) are added to the latter to turn it into a specific HOL formalism. The various notions of equality used in this paper are as follows: When a term is defined (as an abbreviation), the symbol := is used. Primitive equality, written $=^{\tau}$, refers to a logical constant symbol from the HOL language, such that $s_\tau =^{\tau} t_\tau$ is a term of HOL (assuming that s_τ and t_τ are terms, where τ is a type annotation), cf. the details below. Meta equality $\equiv$ denotes set-theoretic identity between objects. Finally, $\equiv_\star$, for $\star \subseteq \{\beta, \eta\}$ is used for syntactic equality modulo $\beta-$, $\eta-$ and $\beta\eta$-conversion, respectively (as in the related work, α-conversion is taken as implicit). The $\beta\eta$-normal form of a term s is written $s^{\downarrow}$.

Syntax of HOL

The set $\mathcal{T}$ of simple types is freely generated from the base types o and ι by juxtaposition. The types o and ι represent the type of Booleans and individuals, respectively. A type $\nu\tau$ represents the type of a total function from objects of type τ to objects of type ν.

Let Σ_τ be a set of constant symbols of type $\tau \in \mathcal{T}$ and let $\Sigma := \bigcup_{\tau\in\mathcal{T}} \Sigma_\tau$ be the union of all typed symbols, called a *signature*. Let further $\mathcal{V}$ denote a set of (typed) variable symbols, with infinitely many variable symbols for each type. From these the terms of HOL are constructed by the following abstract syntax ($\tau, \nu \in \mathcal{T}$):

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V} \mid \; (\lambda X_\tau .\, s_\nu)_{\nu\tau} \; \mid \; (s_{\nu\tau}\; t_\tau)_\nu$$

The terms are called *constants*, *variables*, *abstractions* and *applications*, respectively. The set of all terms of type τ over a signature Σ is denoted $\Lambda_\tau(\Sigma)$, and $\Lambda^c_\tau(\Sigma)$ is used for closed terms, respectively. The notion of free and bound variables are defined as usual, and a term t is called *closed* if t does not contain any free variables.

The type of a term is written as a subscript, but may be dropped if clear from the context (or if not important). Also, parentheses are omitted whenever possible, and application is assumed to be left-associative. Furthermore, the scope of an λ-abstraction's body reaches as far to the right as is consistent with the remaining brackets. Nested applications $s\ t^1\ \ldots\ t^n$ may also be written in vector notation $s\ \overline{t^n}$.

As mentioned earlier, HOL variants in the literature often differ with respect to the choice of primitive logical connectives in their signature Σ. Andrews [4] and Steen [15], for example, assume primitive logical symbols $=_{o\tau\tau}$ in Σ and then consider all other logical connectives as defined terms. Other authors have preferred other choices, and this work is very specific about those choices in each particular context. In any case, $s \neq t$ is used in the remainder as an abbreviation for $\neg(s = t)$. Also, for simplicity, binary logical connectives are written in infix notation; e.g., the term $p_o \vee q_o$ formally represents the application $(\vee_{ooo}\ p_o\ q_o)$. Binder notation is used for universal and existential quantification: The term $\forall X_\tau.\, s_o$ is a short-hand for $\Pi^\tau\ (\lambda X_\tau.\, s_o)$, where Π^τ is a constant symbol denoting universal quantification (more precisely, $\Pi^\tau s_{o\tau}$ is true if and only if $s_{o\tau}$ denotes the full set of objects of type τ). Finally, Leibniz-equality, denoted $\doteq_{o\tau\tau}$, is defined as $\doteq\ := \lambda X_\tau.\, \lambda Y_\tau.\, \forall P_{o\tau}.\, (P\ X) \Rightarrow (P\ Y)$. A Σ-*formula* s_o is a term $s_o \in \Lambda_o(\Sigma)$ of type o and a Σ-*sentence* if it is a closed Σ-formula. The reference to Σ may be omitted if clear from the context.

In the following, variables are denoted by capital letters such as X_τ, Y_τ, Z_τ, and, more specifically, the variable symbols $P_{o\tau}, Q_{o\tau}, P_o, Q_o$ and $F_{\nu\tau}, G_{\nu\tau}$ are used for predicate or Boolean variables and variables of functional type, respectively. Analogously, lower case letters s_τ, t_τ, u_τ denote general terms and $f_{\nu\tau}, g_{\nu\tau}$ are used for terms of functional type.

Semantics of HOL

For an in-depth presentation of the semantics of HOL see the literature, cf. [14, 2, 6, 7, 11]. Here only notions as relevant for this article are reviewed. In particular, the generalised notions of model structures, called applicative structures, as studied by Brown [11] and Benzmüller, Brown and Kohlhase [7] are introduced. This is needed because the key lemmas used in the proof in section §6 are based on these notions.

Generalised notions.

An *applicative structure* is a pair $(\mathcal{D}, @)$ where $\mathcal{D}$ is a typed family of nonempty sets and @ is a collection of application operators $@^{\nu\tau}\colon \mathcal{D}_{\nu\tau} \times \mathcal{D}_\tau \to \mathcal{D}_\nu$ for each function type $\nu\tau \in \mathcal{T}$.

Given an applicative structure, a Σ-*evaluation function* $\mathcal{E}$ is a binary function such that $\mathcal{E}_\varphi(s_\tau) \in \mathcal{D}_\tau$ is the denotation of s under assignment φ, satisfying certain compatibility properties (not discussed here, see [11, Def. 3.2.3]). The structure $(\mathcal{D}, @, \mathcal{E})$ is referred to as Σ-evaluation.

A Σ-*model* is a structure $\mathcal{M} = (\mathcal{D}, @, \mathcal{E}, v)$ such that $(\mathcal{D}, @, \mathcal{E})$ is a Σ-evaluation and $v : \mathcal{D}_o \to \{T, F\}$ is a function such that the logical connectives in Σ have their standard denotation (see [11, Def. 3.3.2]). In this setting, the domain $\mathcal{D}_o$ for formula denotations need not to be binary; the v function grounds the objects in $\mathcal{D}_o$ to a binary evaluation. Finally, a Σ-formula s_o is satisfied by a Σ-model $\mathcal{M} = (\mathcal{D}, @, \mathcal{E}, v)$ and assignment φ if and only if $v(\mathcal{E}_\varphi(s)) = T$.

Standard notions.

The better known notion of general models (also referred to as Henkin models) as defined by Henkin [14] and Andrews [2] is as follows:

A *frame* $\mathcal{D} = \{\mathcal{D}_\tau\}_{\tau\in\mathcal{T}}$ is a collection of nonempty sets called domains, one for each type, such that $\mathcal{D}_o = \{T, F\}$, and $\mathcal{D}_{\nu\tau}$ is some collection of functions with domain $\mathcal{D}_\tau$ and co-domain $\mathcal{D}_\nu$.

An interpretation is a pair $(\mathcal{D}, \mathcal{I})$, where $\mathcal{D}$ is a frame and $\mathcal{I}$ is a function mapping each constant symbol $c_\tau \in \Sigma$ of type τ to its denotation $I(c) \in \mathcal{D}_\tau$. It is assumed that the logical connectives are assigned their standard denotation.

Finally, a general model is a structure $\mathcal{M} = (\mathcal{D}, \mathcal{I}, \mathcal{V})$ such that $(\mathcal{D}, \mathcal{I})$ is an interpretation, and $\mathcal{V}$ is a binary function such that $\mathcal{V}_\varphi(s_\tau) \in \mathcal{D}_\tau$ is the *value* of s_τ with respect to variable assignment φ, satisfying the usual conditions [6]. Furthermore, if $\mathcal{M}$ is a general model, the function $\mathcal{V}$ is uniquely determined. A formula s_o is true in $\mathcal{M}$ under assignment φ if and only if $\mathcal{V}_\varphi(s) = T$.

This version of model structures results from the more general notion above by interpreting the applicative structures $(\mathcal{D}, @)$ as a frame with @ being the standard (set-theoretic) function application, $\mathcal{E}$ being represented by $\mathcal{I}$ (for constant symbols) and by $\mathcal{V}$ (for terms in general), and v taken as identity function.

3 Hintikka sets as defined by Brown

In the formulation of HOL used by Brown [11], the set of primitive logical connectives is not fixed, and can be chosen from the set $\{\top_o, \bot_o, \neg_{oo}, \wedge_{ooo}, \vee_{ooo}, \Rightarrow_{ooo}, \equiv_{ooo}\} \cup \{\Pi_{o(o\tau)} | \tau \in \mathcal{T}\} \cup \{\Sigma_{o(o\tau)} \mid \tau \in \mathcal{T}\} \cup \{=^\tau_{o\tau\tau} | \ \tau \in \mathcal{T}\}$. Brown does discuss *minimal* choices of primitive logical connectives and it is assumed here that every choice considered actually satisfies his minimality criteria. A concrete example of a minimal choice besides Andrews' and Steen's choice of $\{=^\tau_{o\tau\tau} | \ \tau \in \mathcal{T}\}$ would, e.g., be $\{\neg_{oo}, \vee_{ooo}\} \cup \{\Pi_{o(o\tau)} | \tau \in \mathcal{T}\}$. as considered by Benzmüller, Brown and Kohlhase [7, 8].

In this article the notation from Brown [11] is adopted, and the notational conventions as introduced before are used. For example, HOL terms are denoted with lower case symbols s and t (instead of upper case letters as used by Brown), and instead of $\mathit{wff}_\tau(\Sigma)$, which Brown uses to denote the set of HOL terms of type τ, $\Lambda_\tau(\Sigma)$ is used. In the remainder connectives in Brown's variant of HOL are written with a tilde on top, e.g., $\tilde{\neg}$ instead of $\neg$.

Brown distinguishes between so-called internal terms, the elements of $\Lambda_\tau(\Sigma)$, and external propositions (meta-level propositions) in a set $\mathit{prop}(\Sigma)$. External propositions are a technical tool for being able to express sentences even if the signature does not contain sufficient primitive connectives for doing so. This way the impact of different (sparse) signatures on the logic (the set of proofs) can be compared while using a single fixed underlying formalism. Meta-level connectives occurring in external propositions are written with a dot below the connective, e.g., $\underset{.}{\tilde{\neg}}$ for the meta variant of negation $\tilde{\neg}$. They are defined as follows [11, Def. 2.1.20]:[2]

Definition 3.1 (External propositions)**.** Let Σ be a signature. The set $\mathit{prop}(\Sigma)$ of external propositions over Σ is the smallest set satisfying the following clauses (where $\underset{.}{\tilde{=}}$, $\underset{.}{\tilde{\neg}}$, $\underset{.}{\tilde{\vee}}$ and $\underset{.}{\tilde{\forall}}$ are meta-level connectives not part of Σ):

- If $s \in \Lambda_o(\Sigma)$, then $s \in \mathit{prop}(\Sigma)$,
- if $\alpha \in \mathcal{T}$ and $s, t \in \Lambda_\alpha(\Sigma)$, then $[s \underset{.}{\tilde{=}} t] \in \mathit{prop}(\Sigma)$,
- $\underset{.}{\tilde{\top}} \in \mathit{prop}(\Sigma)$,
- if $s \in \mathit{prop}(\Sigma)$, then $[\underset{.}{\tilde{\neg}} s] \in \mathit{prop}(\Sigma)$,
- if $s, t \in \mathit{prop}(\Sigma)$, then $[s \underset{.}{\tilde{\vee}} t] \in \mathit{prop}(\Sigma)$, and
- if $s \in \mathit{prop}(\Sigma)$, then $[\underset{.}{\tilde{\forall}} X_\alpha s] \in \mathit{prop}(\Sigma)$.

Closed propositions $s \in \mathit{prop}(\Sigma)$ are (ambiguously) called sentences; written $s \in \mathit{sent}(\Sigma)$.

Brown introduces the following Hintikka set properties for external propositions [11, Def. 5.5.4]:

Definition 3.2 (Extensional Hintikka sets $\mathcal{H}$)**.** Let $\mathcal{H}$ be a set of external propositions over some signature Σ. Conditions that can be posed on $\mathcal{H}$ are as follows:

[2] Brown's Hintikka set properties are distinguished from Steen's using ▼ signs instead of ∇ (thus deviating from their original naming as used by Brown).

$\vec{\blacktriangledown}_c$ $s \notin \mathcal{H}$ or $\dot{\neg} s \notin \mathcal{H}$.

$\vec{\blacktriangledown}_{\beta\eta}$ If $s \in \mathcal{H}$, then $s^{\downarrow} \in \mathcal{H}$.

$\vec{\blacktriangledown}_{\perp}$ $\dot{\neg}\dot{\top} \notin \mathcal{H}$.

$\vec{\blacktriangledown}_{\neg}$ If $\dot{\neg}\dot{\neg} s \in \mathcal{H}$, then $s \in H$.

$\vec{\blacktriangledown}_{\vee}$ If $s \dot{\vee} t \in \mathcal{H}$, then $s \in \mathcal{H}$ or $t \in \mathcal{H}$.

$\vec{\blacktriangledown}_{\wedge}$ If $\dot{\neg}(s \dot{\vee} t) \in \mathcal{H}$, then $\dot{\neg} s \in \mathcal{H}$ and $\dot{\neg} t \in \mathcal{H}$.

$\vec{\blacktriangledown}_{\forall}$ If $\dot{\forall} X_\tau\, s \in \mathcal{H}$, then $[t/x]s \in \mathcal{H}$ for every closed term $t \in \Lambda^c_\tau(\Sigma)$.

$\vec{\blacktriangledown}_{\exists}$ If $\dot{\neg}(\dot{\forall} X_\tau\, s) \in \mathcal{H}$, then there is a parameter $p_\tau \in \Sigma_\tau$ such that $\dot{\neg}([p/X]s) \in \mathcal{H}$.

$\vec{\blacktriangledown}^{\sharp}$ If $s \in \mathcal{H}$, then $s^{\sharp} \in \mathcal{H}$. Also, if $\dot{\neg} s \in \mathcal{H}$, then $\dot{\neg} s^{\sharp} \in \mathcal{H}$.

$\vec{\blacktriangledown}_m$ If $\dot{\neg}(h\ \overline{s^n}) \in \mathcal{H}$ and $(h\ \overline{t^n}) \in \mathcal{H}$, then there is an i with $1 \leq i \leq n$ such that $\dot{\neg}(s^i \dot{=} t^i) \in \mathcal{H}$.

$\vec{\blacktriangledown}_{dec}$ If p is a parameter and $\dot{\neg}((p\ \overline{s^n}) \dot{=}^{\iota} (p\ \overline{t^n})) \in \mathcal{H}$, then there is an i, $1 \leq i \leq n$, s.t. $\dot{\neg}(s^i \dot{=} t^i) \in \mathcal{H}$.

$\vec{\blacktriangledown}_{\mathfrak{b}}$ If $\dot{\neg}(s \dot{=}^o t) \in \mathcal{H}$, then $\{s, \dot{\neg} t\} \subseteq \mathcal{H}$ or $\{\dot{\neg} s, t\} \subseteq \mathcal{H}$.

$\vec{\blacktriangledown}_{\mathfrak{f}}$ If $\dot{\neg}(f \dot{=}^{\nu\tau} g) \in \mathcal{H}$, then there is a parameter $p_\tau \in \Sigma_\tau$ such that $\dot{\neg}(f\ p \dot{=}^{\nu} g\ p) \in \mathcal{H}$.

$\vec{\blacktriangledown}^{o}_{=}$ If $s \dot{=}^o t \in \mathcal{H}$, then $\{s, t\} \subseteq \mathcal{H}$ or $\{\dot{\neg} s, \dot{\neg} t\} \subseteq \mathcal{H}$.

$\vec{\blacktriangledown}^{\rightarrow}_{=}$ If $f \dot{=}^{\nu\tau} g \in \mathcal{H}$, then $(f\ u \dot{=}^{\nu} g\ u) \in \mathcal{H}$ for every closed term $u \in \Lambda^c_\tau(\Sigma)$.

$\vec{\blacktriangledown}^{r}_{=}$ $\dot{\neg}(s \dot{=}^i s) \notin \mathcal{H}$.

$\vec{\blacktriangledown}^{u}_{=}$ Suppose $(s \dot{=}^i t) \in \mathcal{H}$ and $\dot{\neg}(u \dot{=}^i v) \in \mathcal{H}$. Then $\dot{\neg}(s \dot{=}^i u) \in \mathcal{H}$ or $\dot{\neg}(t \dot{=}^i v) \in \mathcal{H}$. Also, $\dot{\neg}(s \dot{=}^i v) \in \mathcal{H}$ or $\dot{\neg}(t \dot{=}^i u) \in \mathcal{H}$.

A set $\mathcal{H}$ is called an extensional Hintikka set if and only if it satisfies all of the above properties. The collection of all such sets is called $\mathfrak{Hint}_{\beta\mathfrak{f}\mathfrak{b}}(\Sigma)$.

Brown additionally defines the more general notion of elementary Hintikka sets, and thoroughly studies various properties of them [11]. This article focuses on only the extensional variants.

4 Hintikka sets as defined by Steen

In the formulation of HOL, as employed by Steen [15], the equality predicates $=^\tau$, for each type τ, are assumed to be the only logical connectives present in the signature Σ, i.e., $\{=^\tau \mid \tau \in \mathcal{T}\} \subseteq \Sigma$. All (potentially) remaining constant symbols from Σ are called parameters. Such signatures are also referred to as $\Sigma^=$. A formulation of HOL based on equality as sole logical connective originates from Andrew's system $\mathcal{Q}_0$ [3, 4]. Following Andrews [4], further logical connectives are defined as follows (type subscript are dropped in the following whenever possible):[3]

$$
\begin{array}{lcl}
\top_o & := & =^o =^{ooo} =^o \\
\bot_o & := & (\lambda P_o.\, P) =^{oo} (\lambda P_o.\, \top) \\
\neg_{oo} & := & \lambda P_o.\, P =^o \bot \\
\wedge_{ooo} & := & \lambda P_o.\, \lambda Q_o.\, (\lambda F_{ooo}.\, F\ \top\ \top) =^{o(ooo)} (\lambda F_{ooo}.\, F\ P\ Q) \\
\vee_{ooo} & := & \lambda P_o.\, \lambda Q_o.\, \neg\, (\neg P \wedge \neg Q) \\
\Rightarrow_{ooo} & := & \lambda P_o.\, \lambda Q_o.\, \neg P \vee Q \\
\Leftrightarrow_{ooo} & := & \lambda P_o.\, \lambda Q_o.\, P =^o Q \\
\Pi^\tau_{o(o\tau)} & := & \lambda P_{o\tau}.\, P =^{o\tau} \lambda X_\tau.\, \top
\end{array}
$$

Steen uses the following properties for Hintikka sets [15, Def. 3.15]:

Definition 4.1 (Acceptable Hintikka sets $\mathcal{H}$)**.** Let $\mathcal{H}$ be a set of $\Sigma^=$-sentences over some signature $\Sigma^=$ including primitive equality. Conditions that can be posed on $\mathcal{H}$ are as follows:

$\vec{\nabla}_c$ $s \notin \mathcal{H}$ or $\neg s \notin \mathcal{H}$.

$\vec{\nabla}_{\beta\eta}$ If $s \equiv_{\beta\eta} t$ and $s \in \mathcal{H}$, then $t \in \mathcal{H}$.

$\vec{\nabla}^r_=$ $(s \neq^\tau s) \notin \mathcal{H}$, where τ is the type of s.

$\vec{\nabla}^s_=$ If $u[s]_p \in \mathcal{H}$ and $s =^\tau t \in \mathcal{H}$ then $u[t]_p \in \mathcal{H}$, where τ is the type of s and t.

$\vec{\nabla}^+_\mathfrak{b}$ If $s =^o t \in \mathcal{H}$, then $\{s, t\} \subseteq \mathcal{H}$ or $\{\neg s, \neg t\} \subseteq \mathcal{H}$.

$\vec{\nabla}^-_\mathfrak{b}$ If $s \neq^o t \in \mathcal{H}$, then $\{s, \neg t\} \subseteq \mathcal{H}$ or $\{\neg s, t\} \subseteq \mathcal{H}$.

$\vec{\nabla}^+_\mathfrak{f}$ If $f_{\nu\tau} =^{\nu\tau} g_{\nu\tau} \in \mathcal{H}$, then $f\ s = g\ s \in \mathcal{H}$ for any closed term $s \in \Lambda^c_\tau(\Sigma)$.

$\vec{\nabla}^-_\mathfrak{f}$ If $f_{\nu\tau} \neq^{\nu\tau} g_{\nu\tau} \in \mathcal{H}$, then $f\ w \neq g\ w \in \mathcal{H}$ for some parameter $w \in \Sigma_\tau$.

[3] Technically, this formulation is a slight modification of the one used by Andrews [4], since the order of terms in defining equations is swapped in many cases.

$\vec{\nabla}_m$ If s, t are atomic and $s, \neg t \in \mathcal{H}$, then $s \neq^o t \in \mathcal{H}$.

$\vec{\nabla}_d$ If $h\,\overline{s^n} \neq^\tau h\,\overline{t^n} \in \mathcal{H}$, then there is an i with $1 \leq i \leq n$ such that $s^i \neq^{\nu_i} t^i \in \mathcal{H}$, where τ is the type of $h\,\overline{s^n}$, and ν_i is the type of s^i.

A set $\mathcal{H}$ is called an acceptable Hintikka set if and only if it satisfies all of the above properties. The collection of all such sets is called $\mathfrak{H}$.

The type superscript τ of equality connectives $=^\tau$, $\tau \in \mathcal{T}$, is now omitted, too. It is implicitly given by the type of the argument terms, e.g., $s_\tau = t_\tau$ stands for $s_\tau =^\tau t_\tau$; or it is clear from the context.

Definition 4.2. A set $\mathcal{H}$ of formulas is called *saturated* if and only if $s \in \mathcal{H}$ or $\neg s \in \mathcal{H}$ for every closed formula s.

Saturated Hintikka sets allow for a simple construction of models. However, saturatedness is difficult to prove in the context of machine-oriented calculi (in fact, as hard to prove as cut elimination [9]), and hence weaker notions of (unsaturated) Hintikka sets, first proposed by Benzmüller et al. [9], that still admit model existence theorems are used.

4.1 Derived properties

Lemma 4.3 (Basic properties)**.** *Let $\mathcal{H} \in \mathfrak{H}$. Then it holds that*

(a) $\bot \notin \mathcal{H}$

(b) $\neg\top \notin \mathcal{H}$

(c) $\top = \bot \notin \mathcal{H}$

(d) If $s_o = \top \in \mathcal{H}$ or $\top = s_o \in \mathcal{H}$ ($s_o \neq \bot \in \mathcal{H}$ or $\bot \neq s_o \in \mathcal{H}$), then $\{s_o, \top\} \subseteq \mathcal{H}$ ($\{s_o, \neg\bot\} \subseteq \mathcal{H}$)

(e) If $s_o = \bot \in \mathcal{H}$ or $\bot = s_o \in \mathcal{H}$ ($s_o \neq \top \in \mathcal{H}$ or $\top \neq s_o \in \mathcal{H}$), then $\{\neg s_o, \neg\bot\} \subseteq \mathcal{H}$ ($\{\neg s_o, \top\} \subseteq \mathcal{H}$)

(f) If $\neg\bot \in \mathcal{H}$, then $\top \in \mathcal{H}$

(g) If $\top \in \mathcal{H}$, then $\neg\bot \in \mathcal{H}$

(h) If $s = t \in \mathcal{H}$ and $t = u \in \mathcal{H}$, then $s = u \in \mathcal{H}$.

Proof. Let $\mathcal{H} \in \mathfrak{H}$ be an acceptable Hintikka set.

(a) Assume $\bot \in \mathcal{H}$. By definition of $\bot$ it holds $(\lambda P. P) = (\lambda P. \top) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^{+}$ and $\vec{\nabla}_{\beta\eta}$, it follows that $w = \top \in \mathcal{H}$ for any closed term w. Taking $w \equiv \neg\top$ we obtain $\neg\top = \top \in \mathcal{H}$. But then $\vec{\nabla}_{\mathfrak{b}}^{+}$ contradicts $\vec{\nabla}_{c}$. Hence $\bot \notin \mathcal{H}$.

(b) Assume $\neg\top \in \mathcal{H}$. By definition of $\bot$ it holds $(=\neq=) \in \mathcal{H}$, which contradicts $\vec{\nabla}_{=}^{r}$. Hence, $\neg\top \notin \mathcal{H}$.

(c) Assume $\top = \bot \in \mathcal{H}$. Applying $\vec{\nabla}_{\mathfrak{b}}^{+}$ gives us that either $\{\top, \bot\} \subseteq \mathcal{H}$ or $\{\neg\top, \neg\bot\} \subseteq \mathcal{H}$. Either case is impossible by either (a) or (b) of this lemma. Hence, $\top = \bot \notin \mathcal{H}$.

(d) Let $s_o = \top \in \mathcal{H}$ or $\top = s_o \in \mathcal{H}$. In both cases it follows by $\vec{\nabla}_{\mathfrak{b}}^{+}$ that either $\{s, \top\} \subseteq \mathcal{H}$ or $\{\neg s, \neg\top\} \subseteq \mathcal{H}$. Since the latter case contradicts (b) from above, it follows that $\{s, \top\} \subseteq \mathcal{H}$. The negative cases are analogous using $\vec{\nabla}_{\mathfrak{b}}^{-}$.

(e) Let $s_o = \bot \in \mathcal{H}$ or $\bot = s_o \in \mathcal{H}$. In both cases it follows by $\vec{\nabla}_{\mathfrak{b}}^{+}$ that either $\{s, \bot\} \subseteq \mathcal{H}$ or $\{\neg s, \neg\bot\} \subseteq \mathcal{H}$. Since the former case contradicts (a) from above, it follows that $\{\neg s, \neg\bot\} \subseteq \mathcal{H}$. The negative case is analogous using $\vec{\nabla}_{\mathfrak{b}}^{-}$.

(f) Let $\neg\bot \in \mathcal{H}$. Then, by definition of $\bot$, $(\lambda P. P) \neq (\lambda P. \top) \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{f}}^{-}$ and $\vec{\nabla}_{\beta\eta}$ it holds that $p \neq \top \in \mathcal{H}$ for some parameter p. By $\vec{\nabla}_{\mathfrak{b}}^{-}$ it follows that either $\{p, \neg\top\} \subseteq \mathcal{H}$ or $\{\neg p, \top\} \subseteq \mathcal{H}$. Since the former case is ruled out by (b) from above, the latter case yields the desired result.

(g) Let $\top \in \mathcal{H}$, that is, $=^{o} =^{ooo} =^{o} \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{f}}^{+}$ and $\vec{\nabla}_{\beta\eta}$ it follows that $(s_o = t_o) = (s_o = t_o) \in \mathcal{H}$ for every two closed formulas s, t. For $s \equiv t \equiv \neg\bot$ it follows that $(\neg\bot = \neg\bot) = (\neg\bot = \neg\bot) \in \mathcal{H}$, and hence, by $\vec{\nabla}_{\mathfrak{b}}^{+}$, either $\neg\bot = \neg\bot \in \mathcal{H}$ or $\neg\bot \neq \neg\bot \in \mathcal{H}$. Since the latter case is ruled out by $\vec{\nabla}_{=}^{r}$, it follows that $\neg\bot = \neg\bot \in \mathcal{H}$. Again, by $\vec{\nabla}_{\mathfrak{b}}^{+}$, it follows that either $\neg\bot \in \mathcal{H}$ or $\neg(\neg\bot) \in \mathcal{H}$. The latter case is impossible by $\vec{\nabla}_{=}^{r}$ since $\neg(\neg\bot) \equiv (\bot \neq \bot)$ and hence $\neg\bot \in \mathcal{H}$.

(h) Let $s = t \in \mathcal{H}$ and $t = u \in \mathcal{H}$. By $\vec{\nabla}_{=}^{s}$ it follows directly that $s = u \in \mathcal{H}$.

□

Lemma 4.4 (Properties of usual connectives). *Let $\mathcal{H} \in \mathfrak{H}$. Then it holds that*

(a) If $\neg\neg s_o \in \mathcal{H}$, then $s \in \mathcal{H}$

(b) If $(s_o \vee t_o) \in \mathcal{H}$, then $s \in \mathcal{H}$ or $t \in \mathcal{H}$.

(c) If $(s_o \wedge t_o) \in \mathcal{H}$, then $s \in \mathcal{H}$ and $t \in \mathcal{H}$.

(d) If $\Pi^\tau\ s \in \mathcal{H}$, then $s\ t \in \mathcal{H}$ for every closed term t.

(e) If $\neg\Pi^\tau\ s \in \mathcal{H}$, then $\neg(s\ w) \in \mathcal{H}$ for some parameter $w \in \Sigma$.

Proof. Let $\mathcal{H} \in \mathfrak{H}$ be an acceptable Hintikka set.

(a) Let $\neg\neg s_o \in \mathcal{H}$. By definition of $\neg$ and $\vec{\nabla}_{\beta\eta}$ it holds $(s \neq \bot) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{b}}^-$, either $\{s, \neg\bot\} \subseteq \mathcal{H}$, or $\{\neg s, \bot\} \subseteq \mathcal{H}$. As the latter case is impossible by Lemma 4.3(a), it follows that $\{s, \neg\bot\} \subseteq \mathcal{H}$ and, in particular, that $s \in \mathcal{H}$.

(b) Let $s_o \vee t_o \in \mathcal{H}$. By definition of $\vee$, $\neg$ and $\vec{\nabla}_{\beta\eta}$ it holds $\big((\lambda P.\, P \top \top) \neq \lambda P.\, P\ (\neg s)\ (\neg t)\big) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^-$ and $\vec{\nabla}_{\beta\eta}$, it follows that $(p \top \top) \neq (p\ (\neg s)\ (\neg t))$ for some parameter $p \in \Sigma$. By $\vec{\nabla}_d$ either (i) $\top \neq \neg s \in \mathcal{H}$ or (ii) $\top \neq \neg t \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{b}}^-$, applied to both cases, it holds that either (i) $\neg\neg s \in \mathcal{H}$, or (ii) $\neg\neg t \in \mathcal{H}$ (because $\neg\top \notin \mathcal{H}$ by Lemma 4.3(b)). It follows that $s \in \mathcal{H}$ or $t \in \mathcal{H}$ by (a) of this lemma.

(c) Let $s_o \wedge t_o \in \mathcal{H}$. By definition of $\wedge$ and $\vec{\nabla}_{\beta\eta}$ it holds $(\lambda P.\, P\ \top\ \top) = (\lambda P.\, P\ s\ t) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^+$ and $\vec{\nabla}_{\beta\eta}$, it follows that $(w\ \top\ \top) = (w\ s\ t) \in \mathcal{H}$ for every closed term w. By $\vec{\nabla}_{\beta\eta}$, using $w \equiv \lambda x.\, \lambda y.\, x$ and $w \equiv \lambda x.\, \lambda y.\, y$, it holds $\top = s \in \mathcal{H}$ and $\top = t \in \mathcal{H}$, respectively. Application of Lemma 4.3(d) yields the desired result.

(d) Let $\Pi^\tau\ s \in \mathcal{H}$. By definition of Π^τ and $\vec{\nabla}_{\beta\eta}$ it holds $\big(s = \lambda x.\, \top\big) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^+$ and $\vec{\nabla}_{\beta\eta}$, it follows that $s\ t = \top \in \mathcal{H}$ for every closed term t. Application of Lemma 4.3(d) yields the desired result.

(e) Let $\neg\Pi^\tau\ s \in \mathcal{H}$. By definition of Π^τ and $\vec{\nabla}_{\beta\eta}$ it holds that $s \neq (\lambda x.\, \top) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^-$ and $\vec{\nabla}_{\beta\eta}$, it follows that $(s\ p) \neq \top \in \mathcal{H}$ for some parameter p. Application of Lemma 4.3(e) yields the desired result.

□

Leibniz equality $\doteq$ coincides with primitive equality $=$, as is verified next. For practical considerations, however, it is often undesirable to use Leibniz equality for automated equational reasoning [9].

Lemma 4.5 (Properties of Leibniz equality)**.** *Let $\mathcal{H} \in \mathfrak{H}$. Then it holds that*

(a) If $s \doteq t \in \mathcal{H}$, then $s = t \in \mathcal{H}$.

(b) If $\neg(s \doteq t) \in \mathcal{H}$, then $s \neq t \in \mathcal{H}$.

(c) $\neg(s \doteq s) \notin \mathcal{H}$.

(d) If $u[s]_p \in \mathcal{H}$ and $s \doteq t \in \mathcal{H}$, then $u[t]_p \in \mathcal{H}$.

(e) If $s \doteq t \in \mathcal{H}$, then $t \doteq s \in \mathcal{H}$.

(f) If $s \doteq t \in \mathcal{H}$ and $t \doteq u \in \mathcal{H}$, then $s \doteq u \in \mathcal{H}$.

Proof. Let $\mathcal{H} \in \mathfrak{H}$ be an acceptable Hintikka set.

(a) Let $(s \doteq t) \in \mathcal{H}$. By definition of $\doteq$ and $\vec{\nabla}_{\beta\eta}$ it follows that $\big(\lambda P.\, (P\ s) \Rightarrow (P\ t)\big) = \big(\lambda P.\, \top\big) \in \mathcal{H}$, and hence, by $\vec{\nabla}_{\mathfrak{f}}^{+}$ and $\vec{\nabla}_{\beta\eta}$, it holds that $\big((w\ s) \Rightarrow (w\ t)\big) = \top \in \mathcal{H}$ for every closed term w. Then, $(w\ s) \Rightarrow (w\ t) \in \mathcal{H}$ by Lemma 4.3(d). By definition of $\Rightarrow$ and $\vec{\nabla}_{\beta\eta}$ it holds that $\neg(w\ s) \vee (w\ t) \in \mathcal{H}$ and hence, by Lemma 4.4(b), that $\neg(w\ s) \in \mathcal{H}$ or $(w\ t) \in \mathcal{H}$. For $w \equiv (\lambda X.\, s = X)$ it follows by $\vec{\nabla}_{\beta\eta}$ that $\neg(s = s) \in \mathcal{H}$ or $(s = t) \in \mathcal{H}$. Since the former case contradicts $\vec{\nabla}_{=}^{r}$, it follows that $(s = t) \in \mathcal{H}$.

(b) Let $\neg(s \doteq t) \in \mathcal{H}$. By definition of $\doteq$ and $\vec{\nabla}_{\beta\eta}$ it follows that $\big(\lambda P.\, (P\ s) \Rightarrow (P\ t)\big) \neq \big(\lambda P.\, \top\big) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^{-}$ and $\vec{\nabla}_{\beta\eta}$, it holds that $\big((p\ s) \Rightarrow (p\ t)\big) \neq \top \in \mathcal{H}$ for some parameter p. By Lemma 4.3(e) it follows that $\neg\big((p\ s) \Rightarrow (p\ t)\big) \in \mathcal{H}$. Then, by Lemma 4.4(a) and 4.4(c), it follows that $\neg\neg(p\ s) \in \mathcal{H}$ and $\neg(p\ t) \in \mathcal{H}$. Moreover, $(p\ s) \in \mathcal{H}$ by Lemma 4.4(a). By $\vec{\nabla}_m$ it then follows that $(p\ s) \neq (p\ t) \in \mathcal{H}$, and finally, by $\vec{\nabla}_d$, that $s \neq t \in \mathcal{H}$.

(c) Assume $\neg(s \doteq s) \in \mathcal{H}$. By (b) above it follows that $s \neq s \in \mathcal{H}$ which contradicts $\vec{\nabla}_{=}^{r}$. Hence, $\neg(s \doteq s) \notin \mathcal{H}$.

(d) Let $u[s]_p \in \mathcal{H}$ and $s \doteq t \in \mathcal{H}$. By (a) above it holds that $s = t \in \mathcal{H}$ and thus by $\vec{\nabla}_{=}^{s}$ it follows that $u[t]_p \in \mathcal{H}$.

(e) Let $(s \doteq t) \in \mathcal{H}$. By definition of $\doteq$ and $\vec{\nabla}_{\beta\eta}$ it follows that $\big(\lambda P.\, (P\ s) \Rightarrow (P\ t)\big) = \big(\lambda P.\, \top\big) \in \mathcal{H}$. Hence, by $\vec{\nabla}_{\mathfrak{f}}^{+}$ and $\vec{\nabla}_{\beta\eta}$, it holds that $\big((w\ s) \Rightarrow (w\ t)\big) = \top \in \mathcal{H}$ for every closed term w. Then, $(w\ s) \Rightarrow (w\ t) \in \mathcal{H}$ by Lemma 4.3(d). By definition of $\Rightarrow$ and $\vec{\nabla}_{\beta\eta}$ it holds that $\neg(w\ s) \vee (w\ t) \in \mathcal{H}$ and hence by Lemma 4.4(b) that $\neg(w\ s) \in \mathcal{H}$ or $(w\ t) \in \mathcal{H}$. For $w \equiv (\lambda X.\, t \doteq s)$, it follows by $\vec{\nabla}_{\beta\eta}$ that $\neg(t \doteq s) \in \mathcal{H}$ or $(t \doteq s) \in \mathcal{H}$. Assume $\neg(t \doteq s) \in \mathcal{H}$. Since by Lemma 4.5(a) it holds that $s = t \in \mathcal{H}$, it follows by $\vec{\nabla}_{=}^{s}$ that $\neg(t \doteq t) \in \mathcal{H}$, which contradicts (c) above. Hence, $(t \doteq s) \in \mathcal{H}$.

(f) Let $(s \doteq t) \in \mathcal{H}$ and $(t \doteq u) \in \mathcal{H}$. By (d) above it follows that $(s \doteq u) \in \mathcal{H}$.

□

Lemma 4.6 (Sufficient conditions for saturatedness)**.** *Let $\mathcal{H} \in \mathfrak{H}$. It holds that*

(a) If $\top \in \mathcal{H}$, then $\mathcal{H}$ is saturated.

(b) If $\neg s \in \mathcal{H}$ for some closed term s, then $\mathcal{H}$ is saturated.

(c) If $s \vee t \in \mathcal{H}$ for some closed terms s, t, then $\mathcal{H}$ is saturated.

(d) If $s \wedge t \in \mathcal{H}$ for some closed terms s, t, then $\mathcal{H}$ is saturated.

(e) If $\Pi^\tau\ s \in \mathcal{H}$ for some closed term $s_{\tau\to o}$, then $\mathcal{H}$ is saturated.

(f) If $s \doteq t \in \mathcal{H}$ for some closed terms s, t, then $\mathcal{H}$ is saturated.

Proof. Let $\mathcal{H} \in \mathfrak{H}$ be an acceptable Hintikka set.

(a) Let $\top \in \mathcal{H}$, that is, $=^o =^{ooo} =^o \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{f}}^+$ and $\vec{\nabla}_{\beta\eta}$ it follows that $(s_o = t_o) = (s_o = t_o) \in \mathcal{H}$ for every pair of closed formulas s, t. For $s \equiv t \equiv c$ for some closed term c, it follows that $(c = c) = (c = c) \in \mathcal{H}$ and thus, by $\vec{\nabla}_{\mathfrak{b}}^+$ and $\vec{\nabla}_{=}^r$, it holds that $c = c \in H$. By $\vec{\nabla}_{\mathfrak{b}}^+$ it follows that $c \in \mathcal{H}$ or $\neg c \in \mathcal{H}$. Hence, $\mathcal{H}$ is saturated.

(b) If $\neg s \in \mathcal{H}$ for some closed term s, then $s = \bot \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{b}}^+$, it follows that either $\{s, \bot\} \subseteq \mathcal{H}$ or $\{\neg s, \neg\bot\} \subseteq \mathcal{H}$. Since the former case is ruled out by Lemma 4.3(a), it follows that $\neg\bot \in \mathcal{H}$. By Lemma 4.3(f) it follows that $\top \in \mathcal{H}$ and by (a) above it follows that $\mathcal{H}$ is saturated.

(c) If $s \vee t \in \mathcal{H}$ for some closed terms s, t, then by definition of $\vee$ and $\vec{\nabla}_{\beta\eta}$ it follows that $\neg(\neg s \wedge \neg t) \in \mathcal{H}$. An application of (b) yields the desired result.

(d) If $s \wedge t \in \mathcal{H}$ for some closed terms s, t, then by definition of $\wedge$ and $\vec{\nabla}_{\beta\eta}$ it holds $(\lambda g.\, g\ s\ t) = (\lambda g.\, g\ \top\ \top) \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{f}}^+$ and $\vec{\nabla}_{\beta\eta}$ it follows that $s = \top \in \mathcal{H}$ (take $\lambda x.\, \lambda y.\, x$). By $\vec{\nabla}_{\mathfrak{b}}^+$, it follows that either $\{s, \top\} \subseteq \mathcal{H}$ or $\{\neg s, \neg\top\} \subseteq \mathcal{H}$. Since the latter case is ruled out by Lemma 4.3(b), it follows that $\top \in \mathcal{H}$. An application of (a) above yields the desired result.

(e) If $\Pi^\tau\ s \in \mathcal{H}$ for some closed terms s, then by definition of Π^τ and $\vec{\nabla}_{\beta\eta}$ it holds that $s = (\lambda x.\, \top) \in \mathcal{H}$. By $\vec{\nabla}_{\mathfrak{f}}^+$ and $\vec{\nabla}_{\beta\eta}$ it follows that $(s\ w) = \top \in \mathcal{H}$ for every closed term w. By $\vec{\nabla}_{\mathfrak{b}}^+$, it follows that either $\{(s\ w), \top\} \subseteq \mathcal{H}$ or $\{\neg(s\ w), \neg\top\} \subseteq \mathcal{H}$. Since the latter case is ruled out by Lemma 4.3(b), it follows that $\top \in \mathcal{H}$. An application of (a) above yields the desired result.

(f) Let $s \doteq t \in \mathcal{H}$. By definition of Π^τ and $\vec{\nabla}_{\beta\eta}$ it holds that $\Pi(\lambda P.\,(P\ s) \Rightarrow (P\ t)) \in \mathcal{H}$. An application of (e) yields the desired result.

□

Corollary 4.7. *Let $\mathcal{H} \in \mathfrak{H}$ and let $s \neq t \in \mathcal{H}$ or $\neg(s \doteq t) \in \mathcal{H}$ for some closed terms s, t. Then, $\mathcal{H}$ is saturated.*

Proof. As $(s \neq t) \equiv \neg(s = t)$, both cases are a special instance of Lemma 4.6(b). □

Lemma 4.8 (Saturated sets properties)**.** *Let $\mathcal{H} \in \mathfrak{H}$ and let $\mathcal{H}$ be saturated. Then it holds that*

(a) If $s = t \in \mathcal{H}$, then $s \doteq t \in \mathcal{H}$.

(b) If $s = t \in \mathcal{H}$ then $t = s \in \mathcal{H}$.

(c) $s = s \in \mathcal{H}$ for every closed term s.

(d) $s \doteq s \in \mathcal{H}$ for every closed term s.

Proof. Let $\mathcal{H} \in \mathfrak{H}$ and let $\mathcal{H}$ be saturated.

(a) Let $s = t \in \mathcal{H}$ and assume $s \doteq t \notin \mathcal{H}$. Since $\mathcal{H}$ is saturated it follows that $\neg(s \doteq t) \in \mathcal{H}$. Then, by Lemma 4.5(b), it follows that $s \neq t \in \mathcal{H}$, and thus $\{s = t, s \neq t\} \subseteq \mathcal{H}$, which contradicts $\vec{\nabla}_c$. Hence, $s \doteq t \in \mathcal{H}$.

(b) Let $s = t \in \mathcal{H}$ and assume $t = s \notin \mathcal{H}$. Since $\mathcal{H}$ is saturated it follows that $t \neq s \in \mathcal{H}$. Then, by $\vec{\nabla}^s_=$, it follows that $t \neq t \in \mathcal{H}$ which contradicts $\vec{\nabla}^r_=$. Hence, $t = s \in \mathcal{H}$.

(c) Let s be a closed term of some type and assume that $s = s \notin \mathcal{H}$. Since $\mathcal{H}$ is saturated it follows that $s \neq s \in \mathcal{H}$. Since this contradicts $\vec{\nabla}^r_=$ it follows that $s = s \in \mathcal{H}$.

(d) Let s be a closed term of some type and assume that $s \doteq s \notin \mathcal{H}$. Since $\mathcal{H}$ is saturated it follows that $\neg(s \neq s) \in \mathcal{H}$. Since this is impossible by Lemma 4.5(c) it follows that $s \doteq s \in \mathcal{H}$.

□

Lemma 4.9 (Properties of negated equalities)**.** *Let $\mathcal{H} \in \mathfrak{H}$. It holds that*

(a) If $s \neq t \in \mathcal{H}$, then $t \neq s \in \mathcal{H}$.

(b) If $\neg(s \doteq t) \in \mathcal{H}$*, then* $\neg(t \doteq s) \in \mathcal{H}$

(c) If $s \neq t \in \mathcal{H}$*, then* $\neg(s \doteq t) \in \mathcal{H}$*.*

Proof. Let $\mathcal{H} \in \mathfrak{H}$.

(a) Let $s \neq t \in \mathcal{H}$ and assume that $t \neq s \notin \mathcal{H}$. By Corollary 4.7 it follows that $\mathcal{H}$ is saturated and hence $t = s \in \mathcal{H}$. By saturation and Lemma 4.8(b) it follows that $s = t \in \mathcal{H}$, and thus $\{s \neq t, s = t\} \subseteq \mathcal{H}$, which contradicts $\vec{\nabla}_c$. Hence, $t \neq s \in \mathcal{H}$.

(b) Let $\neg(s \doteq t) \in \mathcal{H}$ and assume $\neg(t \doteq s) \notin \mathcal{H}$. By Corollary 4.7 it follows that $\mathcal{H}$ is saturated and hence $t \doteq s \in \mathcal{H}$. Then, by Lemma 4.5(d), it follows that $s \doteq t \in \mathcal{H}$, and thus $\{\neg(s \doteq t), s \doteq t\} \subseteq \mathcal{H}$, which contradicts $\vec{\nabla}_c$. Hence, $\neg(t \doteq s) \in \mathcal{H}$.

(c) Let $s \neq t \in \mathcal{H}$ and assume $\neg(s \doteq t) \notin \mathcal{H}$. By Corollary 4.7 it follows that $\mathcal{H}$ is saturated and hence $(s \doteq t) \in \mathcal{H}$. Then, by Lemma 4.5(a), it follows that $s = t \in \mathcal{H}$, and thus $\{s = t, s \neq t\} \subseteq \mathcal{H}$, which contradicts $\vec{\nabla}_c$. Hence, $\neg(s \doteq t) \in \mathcal{H}$.

□

Definition 4.10 (Leibniz-free)**.** Let S be a set of formulae. S is called *Leibniz-free* if and only if $s \doteq t \notin S$ for any terms s, t.

Corollary 4.11 (Impredicativity Gap)**.** *Let* $\mathcal{H} \in \mathfrak{H}$*.* $\mathcal{H}$ *is saturated or Leibniz-free.*

Proof. Assume that $\mathcal{H}$ is not Leibniz-free. Then there exists some formula $s \doteq t \in \mathcal{H}$. An application of Lemma 4.6(f) yields the desired result. □

4.2 Summary of properties of equality and Leibniz-equality.

The following table contains an overview of the implied properties of $=$ and $\doteq$. A property that holds unconditionally is marked with ✓, a property that holds for saturated Hintikka sets is marked as saturated.

Property	$\star \equiv =$	$\star \equiv \doteq$
$s \star s \in \mathcal{H}$	saturated	saturated
$\neg(s \star s) \notin \mathcal{H}$	✓	✓
If $s \star t \in \mathcal{H}$ and $t \star u \in \mathcal{H}$, then $s \star u \in \mathcal{H}$	✓	✓
If $s \star t \in \mathcal{H}$, then $t \star s \in \mathcal{H}$	saturated	✓
If $u[s]_p \in \mathcal{H}$ and $s \star t \in \mathcal{H}$, then $u[t]_p \in \mathcal{H}$	✓	✓

5 Reduction of $\mathfrak{H}$ (Steen) to $\mathfrak{Hint}_{\beta\mathfrak{fb}}$ (Brown)

In this section the notion of Hintikka sets of Steen is reduced to the notion of Hintikka sets of Brown.

Conceptually, each formula from a set $\mathcal{H} \in \mathfrak{H}$, $\mathcal{H} \subseteq \Lambda_o^c(\Sigma)$, is first translated into its counterpart under a signature $\tilde{\Sigma} \supseteq \{\tilde{\neg}\} \cup \{\tilde{=}^\tau \mid \tau \in \mathcal{T}\}$ containing no other primitive logical connectives. Note that this involves mapping a primitive connective to a primitive connective (i.e., $=$ to $\tilde{=}$) as well as mapping a defined connective to a primitive connective (i.e., $\neg$ to $\tilde{\neg}$). In a second step, the formulae are translated to their meta-counterparts, i.e., to external propositions.[4]

Formally, the mapping is defined as follows: Let $\tilde{\Sigma} := \{\tilde{\neg}\} \cup \{\tilde{=}^\tau \mid \tau \in \mathcal{T}\} \cup \{p \mid p \in \Sigma \text{ is a parameter}\}$. Define $\mathcal{H}_\triangleright := \{(s[\neg \backslash \tilde{\neg}])[= \backslash \tilde{=}] \mid s \in \mathcal{H}\}$, where $s[l \backslash r]$ denotes the term that is obtained by replacing all occurrences of l in s by r. If $s \in \Lambda_o^c(\Sigma)$ then $(s[\neg \backslash \tilde{\neg}])[= \backslash \tilde{=}] \in \Lambda_o^c(\tilde{\Sigma})$, and thus $\mathcal{H}_\triangleright \subseteq \Lambda_o^c(\tilde{\Sigma})$. For any term $s \in \Lambda_o^c(\Sigma)$ the result of this translation applied to s is written as $\widetilde{s}$.

As the Hintikka properties of Brown are defined in terms of external propositions, a set $\mathcal{H}_\triangleright^\sharp \subseteq \mathit{prop}(\tilde{\Sigma})$ is constructed from $\mathcal{H}_\triangleright$ by enriching $\mathcal{H}_\triangleright$ with its external counterparts: To that end, for any term $s_o \in \Lambda_o(\tilde{\Sigma})$, $s^\sharp \in \mathit{prop}(\tilde{\Sigma})$ denotes the term that is constructed by replacing the primitive connective at the head position by its external equivalent,[5] i.e., $(\tilde{\neg} s)^\sharp$ is $\dot{\tilde{\neg}} s$ and $(s \mathrel{\tilde{=}^o} t)^\sharp$ is $s \mathrel{\dot{\tilde{=}}^o} t$. If there is no connective at the head position, the term is left unchanged, i.e., $(p\ \overline{s^n})^\sharp \equiv p\ \overline{s^n}$ and $(X\ \overline{s^n})^\sharp \equiv X\ \overline{s^n}$ whenever $p \in \tilde{\Sigma}$ is a parameter and X is a variable.

Then, $\mathcal{H}_\triangleright^\sharp$ is defined inductively as follows. For $\mathcal{H} \in \mathfrak{H}$, let $\mathcal{H}_\triangleright^\sharp$ be the smallest set of external propositions over $\tilde{\Sigma}$ such that

(1) if $s_o \in \mathcal{H}$, then $\widetilde{s} \in \mathcal{H}_\triangleright^\sharp$, and

(2) if $s_o \in \mathcal{H}$, then $\widetilde{s}^\sharp \in \mathcal{H}_\triangleright^\sharp$, and

(3) if $\dot{\tilde{\neg}} \widetilde{s} \in \mathcal{H}_\triangleright^\sharp$, then $\dot{\tilde{\neg}} \widetilde{s}^\sharp \in \mathcal{H}_\triangleright^\sharp$.

---

[4]In an earlier reduction attempt we tried to reduce Steen's [15] abstract consistency properties to those of Benzmüller, Brown and Kohlhase [7]. This attempt had gaps that could not easily be closed (as was pointed out by an unknown reviewer of [17] and [16]). That mapping replaced primitive equations (in Steen's Hintikka sets) by Leibniz equations (to obtain Hintikka sets in the style of Benzmüller, Brown and Kohlhase). Thereby, it introduced additional universally quantified formulas which in turn triggered the applicability of abstract consistency conditions whose validity could not be ensured by referring to those of Steen. Here we instead map to the structurally better suited, generalised conditions of Brown [11, 10], which enables us to circumvent these earlier problems.

[5]This corresponds to the original definition of $s^\sharp$ by Brown [11, Def. 2.1.38].

Intuitively, this translation takes the translated object-level terms from $\mathcal{H}$ (clause (1)) and additionally expands the set of corresponding meta-level terms by replacing connectives at head positions with their meta-connective equivalents (clause (2)). Connectives directly under a (meta-)negation are also considered (clause (3)).[6] Translations of (deeply) nested connectives to their corresponding meta-level connective are then provided implicitly by the Hintikka closure properties.

Examples of unsaturated Hintikka sets $\mathcal{H}$, $\mathcal{H}_\triangleright$ and $\mathcal{H}^\sharp_\triangleright$ are as follows:

- Let $\mathcal{H} := \{s \mid s \equiv_{\beta\eta} t \text{ for } t \in \{a =^\iota b, a =^\iota a, b =^\iota b, b =^\iota a\}\}$. Then $\mathcal{H}_\triangleright = \{s \mid s \equiv_{\beta\eta} t \text{ for } t \in \{a \tilde{=}^\iota b, a \tilde{=}^\iota a, b \tilde{=}^\iota b, b \tilde{=}^\iota a\}\}$ and $\mathcal{H}^\sharp_\triangleright = \mathcal{H}_\triangleright \cup \{s \mid s \equiv_{\beta\eta} t \text{ for } t \in \{a \underset{\cdot}{\tilde{=}}^\iota b, a \underset{\cdot}{\tilde{=}}^\iota a, b \underset{\cdot}{\tilde{=}}^\iota b, b \underset{\cdot}{\tilde{=}}^\iota a\}\}$.
- Let $\mathcal{H} := \{s \mid s \equiv_{\beta\eta} t \text{ for } t \in \{a =^\iota p(\neg), a =^\iota a, p(\neg) =^\iota p(\neg), p(\neg) =^\iota a\}\}$. Then $\mathcal{H}_\triangleright = \{s \mid s \equiv_{\beta\eta} t \text{ for } t \in \{a \tilde{=}^\iota p(\tilde{\neg}), a \tilde{=}^\iota a, p(\tilde{\neg}) \tilde{=}^\iota p(\tilde{\neg}), p(\tilde{\neg}) \tilde{=}^\iota a\}\}$ and $\mathcal{H}^\sharp_\triangleright = \mathcal{H}_\triangleright \cup \{s \mid s \equiv_{\beta\eta} t \text{ for } t \in \{a \underset{\cdot}{\tilde{=}}^\iota p(\tilde{\neg}), a \underset{\cdot}{\tilde{=}}^\iota a, p(\tilde{\neg}) \underset{\cdot}{\tilde{=}}^\iota p(\tilde{\neg}), p(\tilde{\neg}) \underset{\cdot}{\tilde{=}}^\iota a\}\}$.

It is now established that if $\mathcal{H} \in \mathfrak{H}$ then $\mathcal{H}^\sharp_\triangleright \in \mathfrak{Hint}_{\beta\mathfrak{fb}}$ (i.e., $\mathcal{H}^\sharp_\triangleright$ fulfils all $\vec{\blacktriangledown}$ from §3).

Theorem 5.1 (Reduction of Steen's $\mathfrak{H}$ to Brown's $\mathfrak{Hint}_{\beta\mathfrak{fb}}$)**.** *If $\mathcal{H} \in \mathfrak{H}$, then there exists an extensional Hintikka set $\mathcal{H}' \in \mathfrak{Hint}_{\beta\mathfrak{fb}}$ such that $\mathcal{H}^\sharp_\triangleright \equiv \mathcal{H}'$.*

Proof. Let $\mathcal{H} \in \mathfrak{H}$ be a Hintikka set according to Steen §4, i.e., fulfilling all $\vec{\nabla}$ properties. Each $\vec{\blacktriangledown}$-property of Brown for $\mathcal{H}^\sharp_\triangleright$ is verified individually:

$\vec{\blacktriangledown}_c$ Assume that both $\tilde{s} \in \mathcal{H}^\sharp_\triangleright$ and $\underset{\cdot}{\tilde{\neg}}\tilde{s} \in \mathcal{H}^\sharp_\triangleright$. Then, by definition, $s \in \mathcal{H}$ and $\neg s \in \mathcal{H}$. As this contradicts $\vec{\nabla}_c$, it follows that $\tilde{s} \notin \mathcal{H}^\sharp_\triangleright$ or $\underset{\cdot}{\tilde{\neg}}\tilde{s} \notin \mathcal{H}^\sharp_\triangleright$.

$\vec{\blacktriangledown}_{\beta\eta}$ Let $\tilde{s} \in \mathcal{H}^\sharp_\triangleright$. Then, by definition, $s \in \mathcal{H}$. Since $s \equiv_{\beta\eta} s^\downarrow$ it follows by $\vec{\nabla}_{\beta\eta}$ that $s^\downarrow \in \mathcal{H}$. By definition, it then holds that $\tilde{s}^\downarrow \in \mathcal{H}^\sharp_\triangleright$.

$\vec{\blacktriangledown}_\bot$ $\underset{\cdot}{\tilde{\neg}}\underset{\cdot}{\tilde{\top}} \notin \mathcal{H}^\sharp_\triangleright$ is vacuously true, as $\underset{\cdot}{\tilde{\top}}$ is never the result of any term translation.

$\vec{\blacktriangledown}_\neg$ Let $\underset{\cdot}{\tilde{\neg}}\underset{\cdot}{\tilde{\neg}}\tilde{s} \in \mathcal{H}^\sharp_\triangleright$. Then, by definition, $\neg\neg s \in \mathcal{H}$. By Lemma 4.4(a) it follows that $s \in \mathcal{H}$ and hence $\tilde{s} \in \mathcal{H}^\sharp_\triangleright$.

$\vec{\blacktriangledown}_\vee$ Vacuously true, as $\underset{\cdot}{\tilde{\vee}}$ is never the result of any term translation.

$\vec{\blacktriangledown}_\wedge$ Vacuously true, as $\underset{\cdot}{\tilde{\vee}}$ is never the result of any term translation.

[6] Clauses (2) and (3) intuitively reflect property $\vec{\blacktriangledown}^\sharp$ of Hintikka sets by Brown [11, Def. 5.5.1].

$\vec{\blacktriangledown}_{\forall}$ Vacuously true, as $\tilde{\dot{\forall}}$ is never the result of any term translation.

$\vec{\blacktriangledown}_{\exists}$ Vacuously true, as $\tilde{\dot{\forall}}$ is never the result of any term translation.

$\vec{\blacktriangledown}^{\sharp}$ Holds by definition of $\mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}_{m}$ Let $\tilde{\dot{\neg}}(\tilde{p}\ \overline{\widetilde{s^n}}) \in \mathcal{H}^{\sharp}_{\triangleright}$ and let $(\tilde{p}\ \overline{\widetilde{t^n}}) \in \mathcal{H}^{\sharp}_{\triangleright}$ for some parameter p. Then, by definition, $\neg(p\ \overline{s^n}) \in \mathcal{H}$ and $(p\ \overline{t^n}) \in \mathcal{H}$. By $\vec{\nabla}_m$ it follows that $(p\ \overline{t^n}) \neq (p\ \overline{s^n}) \in \mathcal{H}$. By $\vec{\nabla}_d$ it then follows that there is some i, $1 \leq i \leq n$, such that $t^i \neq s^i \in \mathcal{H}$. By Lemma 4.9(a) it follows that $s^i \neq t^i \in \mathcal{H}$ and hence $\tilde{\dot{\neg}}(\widetilde{s^i} \,\tilde{\dot{=}}\, \widetilde{t^i}) \in \mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}_{dec}$ Let p be a parameter and let $\tilde{\dot{\neg}}((\tilde{p}\ \overline{\widetilde{s^n}}) \,\tilde{\dot{=}}^{\iota}\, (\tilde{p}\ \overline{\widetilde{t^n}})) \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $\neg((p\ \overline{s^n}) =^{\iota} (p\ \overline{t^n})) \in \mathcal{H}$. By $\vec{\nabla}_d$ it follows that there is some i, $1 \leq i \leq n$, such that $\neg(s^i = t^i) \in \mathcal{H}$ and hence $\tilde{\dot{\neg}}(\widetilde{s^i} \,\tilde{\dot{=}}\, \widetilde{t^i}) \in \mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}_{\mathfrak{b}}$ Let $\tilde{\dot{\neg}}(\tilde{s} \,\tilde{\dot{=}}^{o}\, \tilde{t}) \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $\neg(s =^{o} t) \in \mathcal{H}$. By $\vec{\nabla}^{-}_{\mathfrak{b}}$ it follows that $\{s, \neg t\} \subseteq \mathcal{H}$ or $\{\neg s, t\} \subseteq \mathcal{H}$. Hence $\{\tilde{s}, \tilde{\dot{\neg}}\tilde{t}\} \subseteq \mathcal{H}^{\sharp}_{\triangleright}$ or $\{\tilde{\dot{\neg}}\tilde{s}, \tilde{t}\} \subseteq \mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}_{\mathfrak{f}}$ Let $\tilde{\dot{\neg}}(\tilde{f} \,\tilde{\dot{=}}^{\nu\tau}\, \tilde{g}) \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $\neg(f =^{\nu\tau} g) \in \mathcal{H}$. By $\vec{\nabla}^{-}_{\mathfrak{f}}$ it follows that $\neg(f\ p =^{\nu} g\ p) \in \mathcal{H}$ for some parameter p_{τ}. Hence $\tilde{\dot{\neg}}(\tilde{f}\ \tilde{p} \,\tilde{\dot{=}}^{\nu}\, \tilde{g}\ \tilde{p}) \in \mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}^{o}_{=}$ Let $\tilde{s} \,\tilde{\dot{=}}^{o}\, \tilde{t} \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $s =^{o} t \in \mathcal{H}$. By $\vec{\nabla}^{+}_{\mathfrak{b}}$ it follows that $\{s, t\} \subseteq \mathcal{H}$ or $\{\neg s, \neg t\} \subseteq \mathcal{H}$. Hence $\{\tilde{s}, \tilde{t}\} \subseteq \mathcal{H}^{\sharp}_{\triangleright}$ or $\{\tilde{\dot{\neg}}\tilde{s}, \tilde{\dot{\neg}}\tilde{t}\} \subseteq \mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}^{\rightarrow}_{=}$ Let $\tilde{f} \,\tilde{\dot{=}}^{\nu\tau}\, \tilde{g} \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $f =^{\nu\tau} g \in \mathcal{H}$. By $\vec{\nabla}^{+}_{\mathfrak{f}}$ it follows that $f\ s =^{\nu} g\ s \in \mathcal{H}$ for every closed term $s_{\tau} \in \Lambda^{c}_{\tau}(\Sigma)$. Hence $\tilde{f}\ \tilde{s} \,\tilde{\dot{=}}^{\nu}\, \tilde{g}\ \tilde{s} \in \mathcal{H}^{\sharp}_{\triangleright}$ for every closed $\tilde{s}_{\tau} \in \Lambda^{c}_{\tau}(\tilde{\Sigma})$.

$\vec{\blacktriangledown}^{r}_{=}$ Assume $\tilde{\dot{\neg}}(\tilde{s} \,\tilde{\dot{=}}^{i}\, \tilde{s}) \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $\neg(s =^{i} s) \in \mathcal{H}$, contradicting $\vec{\nabla}^{r}_{=}$. Hence $\tilde{\dot{\neg}}(\tilde{s} \,\tilde{\dot{=}}^{i}\, \tilde{s}) \notin \mathcal{H}^{\sharp}_{\triangleright}$.

$\vec{\blacktriangledown}^{u}_{=}$ Let $(\tilde{s} \,\tilde{\dot{=}}^{i}\, \tilde{t}) \in \mathcal{H}^{\sharp}_{\triangleright}$ and $\tilde{\dot{\neg}}(\tilde{u} \,\tilde{\dot{=}}^{i}\, \tilde{v}) \in \mathcal{H}^{\sharp}_{\triangleright}$. Then, by definition, $(s =^{i} t) \in \mathcal{H}$ and $\neg(u =^{i} v) \in \mathcal{H}$. By $\vec{\nabla}_m$ it follows that $(s =^{i} t) \neq^{o} (u =^{i} v) \in \mathcal{H}$. By $\vec{\nabla}_d$ it follows that $s \neq^{i} u \in \mathcal{H}$ or $t \neq^{i} v \in \mathcal{H}$. Hence, $\tilde{\dot{\neg}}(\tilde{s} \tilde{\dot{=}}^{i} \tilde{u}) \in \mathcal{H}^{\sharp}_{\triangleright}$ or $\tilde{\dot{\neg}}(\tilde{t} \tilde{\dot{=}}^{i} \tilde{v}) \in \mathcal{H}^{\sharp}_{\triangleright}$. For the second half, by Corollary 4.7 it is known that $\mathcal{H}$ is saturated and hence by Lemma 4.8(b) it holds that $(t =^{i} s) \in \mathcal{H}$. Applying $\vec{\nabla}^{s}_{=}$ yields $t \neq^{i} u \in \mathcal{H}$ or $s \neq^{i} v \in \mathcal{H}$, and thus $\tilde{\dot{\neg}}(\tilde{t} \,\tilde{\dot{=}}^{i}\, \tilde{u}) \in \mathcal{H}^{\sharp}_{\triangleright}$ or $\tilde{\dot{\neg}}(\tilde{s} \,\tilde{\dot{=}}^{i}\, \tilde{v}) \in \mathcal{H}^{\sharp}_{\triangleright}$.

$\square$

Note that this result technically depends on the exact definitions of logical connectives as introduced by Steen. However, the proofs can be adapted for Andrews' slightly different definitions if desired.

6 Use Case: Bridging Model Existence

In this section, the above reduction is applied to derive a model existence theorem for Steen's Hintikka sets which is, in turn, a key argument for completeness proofs of (machine-oriented) proof calculi.

Informally, the process is as follows: There exists an extensional $\{\tilde{=}, \tilde{\neg}\}$-model $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{bf}}$ such that $\mathcal{M} \models \mathcal{H}^\sharp_\triangleright$. Because $\mathcal{H}^\sharp_\triangleright$ is constructed as an extensional Hintikka set based on negation and equality, the domain of Booleans in $\mathcal{M}$ is known to be bivalent. In order to get a model solely based on equality (as required in the notion of Steen), $\mathcal{M}$ is subsequently restricted to terms over $\{\tilde{=}^\tau \mid \tau \in \mathcal{T}\}$. Finally, an extensional model over frames isomorphic to it is found.[7]

First, important results by Brown [11], used in this reduction, are summarised.

Theorem 6.1 (Model Existence for Extensional Hintikka Sets [11, Thm. 5.7.17])**.** *Let $\mathcal{H} \in \mathfrak{Hint}_{\beta\mathfrak{bf}}(\Sigma)$ be an extensional Σ-Hintikka set. There is an extensional Σ-model $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{bf}}(\Sigma)$ such that $\mathcal{M} \models \mathcal{H}$.*

Theorem 6.2 (Property $\mathfrak{b}$ [11, Theorem 3.3.7])**.** *Let Σ be a signature and $\mathcal{M}$ be an Σ-model. Suppose either $\tilde{\top}, \tilde{\bot} \in \Sigma$ or $\tilde{\neg} \in \Sigma$. Then $\mathcal{M}$ satisfies $\mathfrak{b}$ if and only if $\mathcal{D}_o$ has two elements.*

Theorem 6.3 (Isomorphic Models over Frames [11, Theorem 3.5.6])**.** *Let $\mathcal{M} = (\mathcal{D}, @, \mathcal{E}, v) \in \mathfrak{M}_{\beta\mathfrak{bf}}(\tilde{\Sigma})$ be an extensional Σ-model such that $\mathcal{D}_o$ has two elements. There is an isomorphic Σ-model $\mathcal{M}^h = (\mathcal{D}^h, @^h, \mathcal{E}^h, v^h)$ over frames, in particular $\mathcal{D}^h_o = \{T, F\}$ and v^h is the identity.*

Now a model existence theorem for Steen's Hintikka properties is inferred by bridging to those of Brown: Let $\mathcal{H} \in \mathfrak{H}$ be a Hintikka set (according to Steen) over a signature $\Sigma^=$. Let $\mathcal{H}^\sharp_\triangleright$ be the translated set according to §5. By Theorem 5.1 it follows that $\mathcal{H}^\sharp_\triangleright$ is an extensional $\tilde{\Sigma}$-Hintikka set, for $\tilde{\Sigma} := \{\tilde{\neg}\} \cup \{\tilde{=}^\tau \mid \tau \in \mathcal{T}\} \cup \{p \mid p \in \Sigma^= \text{ is parameter}\}$. By Theorem 6.1 it follows that there is an extensional model

[7] We take a slight indirection here: We first assume negation is part of the translated signature, to make sure there exists an element in $\mathrm{n} \in \mathcal{D}_{oo}$ that is the interpretation of negation. Sadly, it seems there is currently no easier way to enforce its existence; a more convenient way would be to show that there is an extensional model that satisfies $\mathcal{L}_\neg(\mathrm{n})$ without having negation in the signature, cf. [11, 7] for details on these semantic $\mathcal{L}$-properties.

$\mathcal{M} = (\mathcal{D}, @, \mathcal{E}, v) \in \mathfrak{M}_{\beta\mathfrak{bf}}(\tilde{\Sigma})$ such that $\mathcal{M} \models \mathcal{H}^{\sharp}_{\triangleright}$. Since $\tilde{\neg} \in \tilde{\Sigma}$ it follows by Theorem 6.2 that $\mathcal{D}_o$ is bivalent.[8]

Now $\tilde{\neg}$ is eliminated from the signature $\tilde{\Sigma}$ to get an extensional model over $\{\tilde{=}^{\tau} \mid \tau \in \mathcal{T}\}$; this signature is referred to as $\widetilde{\Sigma^{=}}$, i.e., let $\widetilde{\Sigma^{=}} := \tilde{\Sigma} \setminus \{\tilde{\neg}\}$. To that end, let $\mathcal{M}' := (\mathcal{D}, @, \mathcal{E}|_{\Lambda(\widetilde{\Sigma^{=}})}, v)$. $\mathcal{M}'$ is an extensional $\widetilde{\Sigma^{=}}$-model, in particular $\mathcal{D}_o$ is bivalent [11, Theorem 3.3.15]. Now by Theorem 6.3 it follows that there is an $\widetilde{\Sigma^{=}}$-model over frames $\mathcal{M}^h = (\mathcal{D}^h, @^h, \mathcal{E}^h, v^h)$ isomorphic to $\mathcal{M}'$, in particular v^h is the identity function.

The desired Henkin model (in the usual definition style) $\mathcal{M}_{\triangleleft}$ for $\mathcal{H}$ over $\Sigma^{=}$ is constructed as follows: $\mathcal{M}_{\triangleleft} := (\mathcal{D}_{\triangleleft}, \mathcal{I}_{\triangleleft}, \mathcal{V})$, where

- $\mathcal{D}_{\triangleleft} := \mathcal{D}^h$,
- $\mathcal{I}_{\triangleleft} := c \mapsto \mathcal{E}^h(\widetilde{c})$ for all $c \in \Sigma^{=}$, and
- $\mathcal{V} := \mathcal{E}^h$.

It is simple to check that $\mathcal{M}_{\triangleleft}$ is a Henkin model; in particular, $\mathcal{I}(=^{\tau})(\mathsf{a}, \mathsf{b}) = T$ if and only if $\mathsf{a} \equiv \mathsf{b}$ for every $\mathsf{a}, \mathsf{b} \in \mathcal{D}^h_{\tau}$, $\tau \in \mathcal{T}$. This is because it is ensured in $\mathcal{M}^h$ by definition that for $\mathsf{q} \equiv \mathcal{E}^h(\tilde{=}^{\tau})$ it holds that $v^h(\mathsf{q}@^h a @^h b) = T$ if and only if $a \equiv b$ (the respective property is called $\mathcal{L}_{=^{\tau}}(\mathsf{q})$ in [11, Def. 3.3.1]).

Finally, it is necessary to verify that $\mathcal{M}_{\triangleleft} \models \mathcal{H}$ holds. Let $s_o \in \mathcal{H}$. By definition $\mathcal{M}_{\triangleleft} \models s$ if and only if $\|s\|^{\mathcal{M}_{\triangleleft}, g} \equiv T$ for every g. An induction over the structure of s yields the desired result: If s is an equality of the form $(l =^{\tau} r)$, then $\|l =^{\tau} r\|^{\mathcal{M}_{\triangleleft}, g} \equiv T$ if and only if $\|l\|^{\mathcal{M}_{\triangleleft}, g} \equiv \|r\|^{\mathcal{M}_{\triangleleft}, g}$. Since $\widetilde{l} \tilde{=}^{\tau} \widetilde{r} \in \mathcal{H}^{\sharp}_{\triangleright}$, it is known that $\mathcal{M} \models \widetilde{l} \tilde{=}^{\tau} \widetilde{r}$ and, consequently, that $\mathcal{E}_{\varphi}(\widetilde{l}) \equiv \mathcal{E}_{\varphi}(\widetilde{r})$ for every assignment φ. It follows that $\mathcal{E}^h_{\varphi}(l[=\backslash\tilde{=}]) \equiv \mathcal{E}^h_{\varphi}(r[=\backslash\tilde{=}])$, and hence, by definition of $\mathcal{M}_{\triangleleft}$ and the induction hypothesis, $\mathcal{I}_{\triangleleft}(=^{\tau})(\|r\|^{\mathcal{M}_{\triangleleft}, g}, \|r\|^{\mathcal{M}_{\triangleleft}, g}) = T$ and thus $\|l =^{\tau} r\|^{\mathcal{M}_{\triangleleft}, g} \equiv T$, for every g. For parameters $p \in \Sigma^{=}$, the proposition follows directly. For complex formulas of the form $(s\ \overline{s^n})$ and for abstractions, the induction hypotheses is applied to every sub-formula. This construction yields:

Theorem 6.4 (Bridged Model Existence)**.** *Let $\mathcal{H} \in \mathfrak{H}$ be a $\Sigma^{=}$-Hintikka set. Then there exists a $\Sigma^{=}$-Henkin model $\mathcal{M}$ such that $\mathcal{M} \models \mathcal{H}$.*

As intended, the above model existence result for Steen's $\Sigma^{=}$-Hintikka sets can be established by reduction to earlier results of Brown [11], and this result can be used to prove the completeness of the higher order extensional paramodulation calculus underlying the theorem proving system Leo-III [17].

[8]The fact that $\mathcal{M}$ satisfies property $\mathfrak{b}$ follows directly from the fact that $\mathcal{M} \in \mathfrak{M}_{\beta\mathfrak{bf}}(\tilde{\Sigma})$.

References

[1] Peter B. Andrews. Resolution in type theory. *J. Symb. Log.*, 36(3):414–432, 1971.

[2] Peter B. Andrews. General models and extensionality. *J. Symb. Log.*, 37(2):395–397, 1972.

[3] Peter B. Andrews. General models, descriptions, and choice in type theory. *J. Symb. Log.*, 37(2):385–394, 1972.

[4] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory.* Applied Logic Series. Springer, 2002.

[5] Hendrik Pieter Barendregt, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types.* Perspectives in logic. Cambridge University Press, 2013.

[6] Christoph Benzmüller and Peter Andrews. Church's Type Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, pages 1–62 (in pdf version). Metaphysics Research Lab, Stanford University, summer 2019 edition, 2019.

[7] Christoph Benzmüller, Chad Brown, and Michael Kohlhase. Higher-Order Semantics and Extensionality. *J. Symb. Log.*, 69(4):1027–1088, 2004.

[8] Christoph Benzmüller, Chad Brown, and Michael Kohlhase. Semantic Techniques for Cut-Elimination in Higher-Order Logics. SEKI Report SR-2004-07, Saarland University, Saarbrücken, Germany, 2004.

[9] Christoph Benzmüller, Chad Brown, and Michael Kohlhase. Cut-simulation and impredicativity. *Logical Methods in Computer Science*, 5(1:6):1–21, 2009.

[10] Chad E. Brown. *Set Comprehension in Church's Type Theory.* PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University, 2004.

[11] Chad E. Brown. *Automated Reasoning in Higher-Order Logic. Set Comprehension and Extensionality in Church's Type Theory*, volume 10 of *Studies in Logic.* College Publications, 2007.

[12] Alonzo Church. A formulation of the simple theory of types. *J. Symb. Log.*, 5(2):56–68, 1940.

[13] Alonzo Church. *The Calculi of Lambda Conversion*, volume 6 of *Annals of Mathematics Studies.* Princeton University Press, 1941.

[14] Leon Henkin. Completeness in the theory of types. *J. Symb. Log.*, 15(2):81–91, 1950.

[15] Alexander Steen. *Extensional paramodulation for higher-order logic and its effective implementation Leo-III.* PhD thesis, Freie Universität Berlin, Berlin-Dahlem, Germany, 2018. Published in series DISKI – Dissertations in Artificial Intelligence, Vol. 345, AKA Verlag and IOS Press, 2018. ISBN 978-1-61499-919-5.

[16] Alexander Steen and Christoph Benzmüller. On reductions of Hintikka sets for higher-order logic. *arXiv:2004.07506*, pages 1–10, 2020.

[17] Alexander Steen and Christoph Benzmüller. Extensional higher-order paramodulation in Leo-III. *J. Autom. Reason.*, 65(6):775–807, 2021.

 Received 19 April 2024

Effects of the Strict-Tolerant Approach on Constructive Logics

Victor Barroso-Nascimento*
University College London
victorluisbn@gmail.com

German Lourenço Mejia†
Federal Institute of São Paulo
germanufrj@gmail.com

Abstract

This paper extends the literature on the strict-tolerant logical approach by applying its methods to intuitionistic and minimal logic. In short, the strict-tolerant approach modifies the usual notion of logical consequence by stipulating that, in order for an inference to be valid, from the truth of the premises must follow the non-falsity of the conclusion. This notion can also be generalized to define strict-tolerant metainferences, metametainferences and so on, which may or may not generate logics distinct from those obtained on the inferential level. It is already known that strict-tolerant definitions can make the notion of inference for non-classical logics collapse into the classical notion, but the strength of this effect is not yet fully known. This paper shows that intuitionistic strict-tolerant inferences also collapse into classical ones, but minimal ones do not. However, minimal strict-tolerant logic has the property that no inferences are valid (which is not carried over to the metainferential level). Additionally, it is shown that the logics obtained from intuitionistic, minimal and classical logic at at the metainferential level are distinct from each other.

The authors would like to thank Luiz Carlos Pereira for his many comments and suggestions in a preliminary version of this paper, as well as for the enthusiastic encouragement concerning its writing. They would also like to thank the anonymous reviewers for their fruitful suggestions and Eduardo Barrio for sending the invitation to a private seminar on Strict-Tolerant logics.

*Barroso-Nascimento is supported by the Leverhulme grant RPG-2024–196. He was also supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

†Lourenço Mejia is supported by the Federal Instutitue of São Paulo, campus of Barretos.

1 Introduction

The main goal of this paper is to further the study of non-classical logics by investigating some relations between criterias for inferential validity and constructivism. Specifically, we present technical results concerning a combination of the strict-tolerant approach [2] with intuitionistic and minimal logic. It is claimed that, aside from making evident some perils of combining strict-tolerant notions of consequence with paraconsistent and constructive base logics, these results provide partial answers to questions raised by Fitting at [14, pg. 393] of (i) whether there is a strict-tolerant version of intuitionistic logic and (ii) what would happen if we were to define a strict-tolerant logic in which inferences are valid whenever from the intuitionistic truth of the premises follows the classical truth of one of the conclusions.

It is worthy of note that the negative part of our results only apply to some straightforward adaptations of the Strict-Tolerant framework. Fitting himself has recently shown [15] that more nuanced definitions yield logics that have the same relation to intuitionistic logic than traditional Strict-Tolerant logic has to classical logic, meaning that our negative result only apply under certain conditions. This has, of course, no bearing on our positive results (on differences between our logics at the metainferential level), which remain of independent interest.

Our discussion is structured as follows. In the second section, we briefly comment on the relationship between the study of inferences and constructivism, providing some context for our technical results. In the third section, we offer the basic definitions required for our proofs. In the fourth section, we present this paper's main technical results. In the fifth and final section, we conclude by providing some brief comments on the results of section four, their perceived inner workings and some possible relations to future research.

2 The inferential justification of constructivism

Argumentative practices implicitly rely on the possibility of justifying claims by providing reasons for them. In the context of an argument, the act of drawing a conclusion from something purportedly providing it with a justification is called an *inference*. Inferences are called *valid* when they succeed in providing acceptable justifications and *invalid* when they fail to do so. Since arguments themselves are structured as successive acts of inference, the acceptability of an argument depends on the acceptability of its inferences, so an argument is *valid* whenever all its inferences are valid and *invalid* when at least one is not.

Although the systematic study of argumentation has integrated logic since the works of Aristotle [31], there is still no consensus on what makes inferences, and therefore arguments, valid. In fact, Prawitz singles this as the most fundamental question of General Proof Theory [33], the branch of logic dedicated to study of argumentative structures. It is also fundamental for philosophical doctrines in which inferences play a prominent role – such as *inferentialism*, according to which the contents of propositions are exhaustively determined by their possible uses as premises and conclusions of inferences [8].

The study of inferences and inferential validity is of particular interest in the context of constructive logics, for which inferentialism is contemporarily viewed as one of the most robust philosophical justifications. Michael Dummett and Dag Prawitz in particular have ostensively argued that inferences and the role they play in the determination of propositional meaning provides both a justification for the laws of intuitionistic logic and motives for rejecting classical canons of reasoning [11, pgs. 245-279][10][27][29][28]. Also worthy of note is the remarkable affinity between formal semantics defined using the concept of inference and intuitionistic logic [40][38][42][17], although under suitable conditions the framework can also yield semantics for classical logic [37][16][6].

When it comes to logical consequence, traditional accounts establish truth-preservation, and sometimes relevance, as the main ingredient of validity [36]. In such views, a valid argument must at least be capable of guaranteeing that from the truth of its premises follows the truth of its conclusion. Despite its simplicity, the feasibility of this requirement remains highly contentious [24]. It is also regarded as especially problematic by constructivists, as the very concept of truth – unless given unorthodox epistemic readings instead of the traditional ontological one – is considered inadequate for semantical analysis, either in general or specifically in the context of mathematics [30][23][12].

The use of different criteria for validity has been suggested in the literature, often as a direct answer to problems arising from the use of truth preservation. In particular, the literature on *strict-tolerant logics* suggests that by adopting a *weaker* criterion which submits premises to a "strict" evaluation standard and conclusions to a "tolerant" one it becomes possible to satisfactorily deal with problems pertaining to semantic paradoxes [9]. The approach stipulates that an inference is valid whenever from the truth of the premises follows at least the non-falsity of the conclusion, which is the same as truth-preservation in classical logic but yields a distinct entailment notion in other contexts [2][14].

As shown in [1] and [4], this approach allows one to deal with paradoxes related to vagueness by semantically defining sequent-based logics in which the *Cut* principle (transitivity of deduction) does not hold in general. Such systems easily allow one to provide interesting calculi for paraconsistent logics such as the Logic of Paradox [34], whose affinity with the strict-tolerant approach had already been noted before

[2]. The literature also shows that, by applying the approach in a paraconsistent setting, it becomes possible to propose unorthodox solutions to traditional problems: in [3], for instance, an hierarchy of paraconsistent logics formulated using the strict-tolerant approach is used to propose an entirely new criterion of identity between logics, although the philosophical significance of this hierarchy is challenged in [20].

In light of the importance given to inferences in the philosophy of intuitionism, the investigation of how deviant concepts of inferential validity fare in constructive frameworks seems to be warranted. This paper will specifically investigate the inferential and metainferential behavior of strict-tolerant notions of validity in Kripke semantics for minimal and intuitionistic logic, the former being a paraconsistent version of the latter which arose from minor disagreements concerning the possibility of constructively justifying the principle of *ex falso sequitur quodlibet* [19, pg. 102][21][43]. The inclusion of minimal logic is justified not only by the proximity between both logics and their respective philosophical justifications, but also by the aforementioned well-known connection between paraconsistent logics and the strict-tolerant approach.

3 Basic definitions

We start by defining semantics for intuitionistic and minimal logic, using the notation and definitions of [35] and taking some elements from [25].

Definition 3.1. *The language L of minimal and intuitionistic logic is comprised of atomic formulas, the unary logical connective $\bot$ and the binary logical connectives $\wedge$, $\vee$ and $\rightarrow$.*

Convention 3.2. *$\neg A$ is used as an abbreviation for $A \rightarrow \bot$.*

Definition 3.3. *A interpretation is a triple (W, R, v), in which W is a non-empty set of worlds w, R is a reflexive and transitive relation on W and v is a function assigning one of two values $\langle 1, 0\rangle$ to pairs $\langle A, w\rangle$ comprised of formulas A and worlds w.*

Convention 3.4. *$v_w(A) = 1$ is used to express that the function v assigns value 1 to the pair $\langle A, w\rangle$, and $v_w(A) = 0$ to express the assignment of 0 to the same pair.*

Definition 3.5. *A minimal interpretation is an interpretation in which the following constraint is satisfied: for all w and w' in W, if $v_w(A) = 1$ and $wRw', v_{w'}(A) = 1$.*

Definition 3.6. *A intuitionistic interpretation is a minimal interpretation in which $v_w(\bot) = 0$ for all w.*

Definition 3.7. *A classical interpretation is a intuitionistic interpretation such that, for all w and w' in W and all formulas A, $v_w(A) = v_{w'}(A)$.*

Definition 3.8. *The semantic clauses for molecular formulas in interpretations are as follows:*

1. $v_w(A \wedge B) = 1$ *if and only if* $v_w(A) = 1$ *and* $v_w(B) = 1$.
2. $v_w(A \vee B) = 1$ *if and only if* $v_w(A) = 1$ *or* $v_w(B) = 1$.
3. $v_w(A \to B) = 1$ *if for all w' such that wRw',* $v_{w'}(A) = 0$ *or* $v_{w'}(B) = 1$

Notice that the accessibility relation R becomes useless in classical interpretations, as the preservation of values across worlds boils the implicational clause down to making an implication true whenever the antecedent is false in all worlds or the consequent is true in all worlds (and thus, since the values of implications and conjunctions are also constant across worlds, the semantic collapses into usual two-valued Boolean truth functions).

Definition 3.9. *Truth and falsity in an interpretation can be defined as followed:*

1. *A is true in an interpretation* $\langle W, R, v\rangle$ *iff* $v_w(A) = 1$ *for all* $w \in W$*;*
2. *A is false in an interpretation* $\langle W, R, v\rangle$ *iff* $v_w(\neg A) = 1$ *for all* $w \in W$*;*

Three things should be noted about this definition:

(I) We are adopting a strong notion of falsity, according to which a formula is false if and only if its negation is true. This is intended to be a *constructive* notion of falsity, so the falsity of a formula is only recognized when we have an actual refutation of it (and thus we are also able to assert its negation). Therefore, a non-true formula is not automatically false, which is very much in line with the spirit of intuitionistic and minimal logic;

(II) In accordance to what is usually expected of constructive theories, a formula can be neither true nor false in intuitionistic and minimal interpretations;

(III) In intuitionistic logic, falsity of A is equivalent to $v_w(A) = 0$ for all $w \in W$, which makes it so that a formula cannot be both true and false. This is not the case in minimal logic, which admits simultaneous truth and falsity (an unsurprising feature for a logic with paraconsistent characteristics). Another plausible definition for falsity would be to directly use the condition $v_w(A) = 0$ for all $w \in W$, but an unpleasant consequence of this would be that proofs of $\neg A$ would no longer establish the falsity of A in minimal logic – and thus this definition would change the meaning usually assigned to negation.

Definition 3.10. $\Gamma \vDash_m A$ *holds if every minimal interpretation which makes all formulas in* Γ *true also makes the formula* A *true.*

Definition 3.11. $\Gamma \vDash_i A$ *holds if every intuitionistic interpretation makes all formulas in* Γ *true also makes the formula* A *true.*

Definition 3.12. $\Gamma \vDash_c A$ *holds if every classical interpretation which makes all formulas in* Γ *true also makes the formula* A *true.*

We now proceed to the definition of strict-tolerant variants of those logics, drawing from the definitions in [2] and [3]:

Definition 3.13. *A strict-tolerant inference* $\Gamma \Rightarrow \Delta$ *holds if every interpretation which makes all formulas in* Γ *true do not make all formulas in* Δ *false.*

Corollary 3.14. *Let* $\Delta = \{A_1, ..., A_n\}$ *for non-empty* Δ*. A strict-tolerant inference* $\Gamma \Rightarrow \Delta$ *holds under classical interpretations if and only if* $\Gamma \vDash_c A$ *holds, in which* A *is the disjunction* $A_1 \vee ... \vee A_n$*.*

Corollary 3.14 follows directly from classical bivalence, Definition 3.13 and the classical definition of disjunction. It shows that, unless we are interested in inferences with empty succedents, we need only consider classical inferences with at most a single formula on the succedent, as non-empty sets can be replaced by a single disjunctions without any loss.

The same can be shown for intuitionistic and minimal inferences:

Theorem 3.15. *Let* $\Delta = \{A_1, ..., A_n\}$ *for non-empty* Δ*. A strict-tolerant inference* $\Gamma \Rightarrow \Delta$ *holds under minimal (or intuitionistic) interpretations if and only if* $\Gamma \Rightarrow A$ *holds, in which* A *is the disjunction* $A_1 \vee ... \vee A_n$*.*

Proof. For the left-to-right direction, let $\Gamma \Rightarrow \Delta$ hold. Then, by Definitions 3.13 and 3.9, for every interpretation which makes Γ true we have $v_w(\neg A_k) = 0$ for some $A_k \in \Delta$ and some w of the interpretation. For this w it holds that there is a w' with wRw' such that $v_{w'}(A_k) = 1$ and $v_{w'}(\bot) = 0$. From $v_{w'}(A_k) = 1$ we can iterate the disjunction semantic clause to get $v_{w'}(A_1 \vee ...A_k... \vee A_n) = 1$, and since $v_{w'}(\bot) = 0$ we have $v_{w'}(\neg(A_1 \vee ...A_k... \vee A_n)) = 0$, which prevents $A_1 \vee .... \vee A_n$ from being false in the interpretation and thus establishes $\Gamma \vDash A$, as the result was proven for arbitrary interpretations which make Γ true.

For the right-to-left direction, let $\Gamma \vDash A$. Then for every interpretation which makes Γ true we have $v_w(\neg(A_1 \vee .... \vee A_n)) = 0$ for some w. Again, for this w we have a w' with wRw' in which $v_{w'}(A_1 \vee ... \vee A_n) = 1$ and $v_{w'}(\bot) = 0$. By repeatedly decomposing the disjunction we have $v_{w'}(A_k) = 1$ for some $A_k \in \Delta$, and since $v_{w'}(\bot) = 0$ we have $v_{w'}(\neg A_k) = 0$, which prevents A_k from being false. Since this can be done for any interpretation which makes Γ true, we have $\Gamma \Rightarrow \Delta$. □

In order to simplify some of the proofs, from now on we implicitly use these results and consider only inferences with a single formula on the succedent.

Definition 3.16. *The strict-tolerant intuitionistic consequence relation* $\Gamma \vDash^i_{ST} A$ *holds if* $\Gamma \Rightarrow A$ *holds for all intuitionistic interpretations.*

Definition 3.17. *The strict-tolerant minimal consequence relation* $\Gamma \vDash^m_{ST} A$ *holds if* $\Gamma \Rightarrow A$ *holds for all minimal interpretations.*

We will now define the notion of *strict-tolerant metainference.* A strict-tolerant metainference is a higher-order inference which has a (possibly empty) list of strict-tolerant inferences as its premises and a single strict-tolerant inference as its conclusions. We denote this relation by $\Rightarrow^*$, and define it as follows:

Definition 3.18. *A strict-tolerant metainference* $(\Gamma_1 \Rightarrow A_1), ..., (\Gamma_n \Rightarrow A_n) \Rightarrow^* (\Gamma_{n+1} \Rightarrow A_{n+1})$ *holds if every interpretation either (i) makes* Γ_k *true and* A_k *false for some* k $(1 \leqslant k \leqslant n)$*, (ii) does not make* Γ_{n+1} *true or (iii) makes* A_{n+1} *not false.*

In more intuitive but less practical terms, for a metainference to hold it must be the case that in every interpretation which makes Γ_k true and A_k not false (for all Γ_k and A_k on the antecedent, if any), if this interpretation makes Γ_{n+1} true, it also makes A_{n+1} not false. Notice also that condition (i) cannot be satisfied when the list of inferences on the antecedent is empty, which forces us to consider clauses (ii) and (iii) for all interpretations and thus makes validity of the metainference equivalent to validity of the inference on the succedent (compare clauses (ii) and (iii) with Definition 3.13).

Definition 3.19. *Let* Θ *be a (possibly empty) sequence of strict-tolerant inferences and* S *be a strict-tolerant inference. Then, the strict-tolerant minimal metaconsequence* $\Theta \vDash^m_{STM} S$ *holds if the metainference* $\Theta \Rightarrow^* S$ *holds in all minimal interpretations.*

Definition 3.20. *Let* Θ *be a (possibly empty) sequence of strict-tolerant inferences and* S *be a strict-tolerant inference. Then, the strict-tolerant intuitionistic metaconsequence* $\Theta \vDash^i_{STM} S$ *holds if the metainference* $\Theta \Rightarrow^* S$ *holds in all intuitionistic interpretations.*

Definition 3.21. *Let Θ be a (possibly empty) sequence of strict-tolerant inferences and S be a strict-tolerant inference. Then, the strict-tolerant classical metaconsequence $\Theta \vDash^c_{STM} S$ holds if the metainference $\Theta \Rightarrow^* S$ holds in all classical interpretations.*

Naturally, the notion of metainference can be generalized so as to create an hierarchy of logics, as done in [3]. However, as will be briefly shown, the inferential and meta-inferential levels are already sufficient to distinguish between classical, intuitionistic and minimal logics with regards to identity, so the meta-inferential level is sufficient for the purposes of this paper.

4 Main results

We now prove some preparatory lemmata and then proceed to prove this paper's main results.

Lemma 4.1 (Glivenko's Theorem, Generalized)**.** *If $\Gamma \vDash_c A$, then $\Gamma \vDash_i \neg\neg A$.*

Proof. The standard proof can be seen in Proposition 2 of [26], which shows that $\Gamma \vDash_c A$ implies $\Gamma, \neg A \vDash_i \bot$ and allows us to conclude Lemma 4.1 through the deduction theorem for intuitionistic logic. □

This is a straightforward generalization of Glivenko's original theorem, which was proved only for empty Γ [18]. It essentially relies on the fact, when A is a classical but not intuitionistic validity, every application of *reductio ad absurdum* in the proof of A can be transformed into a proof of negation with conclusion $\neg\neg A$. It may therefore also be viewed as a corollary of Seldin's normalization strategy, since it shows that if there is a classical proof of A then there is a classical proof of A containing only one application of *reductio ad absurdum*, which always appears at the very last step [41].

The result can be extended to the fragment of minimal logic without implication [13]. Surprisingly, if one does include implication in the language it no longer holds:

Lemma 4.2. *There is a formula A such that $\vDash_c A$ but $\nvDash_m \neg\neg A$.*

Corollary 4.3. *There is a set Γ and a formula A such that $\Gamma \vDash_c A$ but $\Gamma \nvDash_m \neg\neg A$.*

Proof. We can obtain a quick semantic counterexample by considering an instance $\neg a \to (a \to b)$, with atomic a and b, of the classical tautology $\neg A \to (A \to B)$. Let I be minimal interpretation with only two worlds (w and w') such that:

(I) wRw' and the relations obtained through reflexivity and transitivity of R hold, but not $w'Rw$;

(II) $v_w(a) = 1$ and $v_w(b) = v_w(\bot) = 0$;

(III) $v_{w'}(a) = v_{w'}(\bot) = 1$ and $v_{w'}(b) = 0$.

By Convention 3.2, clause 3 of Definition 3.8 and our choice of valuation function we have both $v_w(\neg a) = 0$ and $v_{w'}(\neg a) = 1$.

Since $v_{w'}(a) = 1$, $v_{w'}(b) = 0$ and w' is related to itself via reflexitivity of R, the semantic clauses yield $v_{w'}(a \to b) = 0$, which together with $v_{w'}(\neg a) = 1$ yields $v_{w'}(\neg a \to (a \to b)) = 0$. However, since $v_{w'}(\bot) = 1$, we have both $v_{w'}(\neg(\neg a \to (a \to b))) = 1$ and $v_{w'}(\neg\neg(\neg a \to (a \to b))) = 1$.

Now, since wRw', we have $v_w(\neg a \to (a \to b)) = 0$ due to the valuation of $\neg a$ and $a \to b$ in w'. Since $v_w(\neg a \to (a \to b)) = 0$ and $v_{w'}(\bot) = 1$, as no other world is accessible from w, by clause 3 of definition 6 we also have $v_w(\neg(\neg a \to (a \to b))) = 1$. However, since $v_w(\neg(\neg a \to (a \to b))) = 1$ and $v_w(\bot) = 0$, we have $v_w(\neg\neg(\neg a \to (a \to b))) = 0$, which makes it so that $\neg\neg(\neg a \to (a \to b))$ is not true in this interpretation (according to Definition 3.9), and so $\not\models_m \neg\neg(\neg a \to (a \to b))$ by Definition 3.10. □

We will now proceed to prove the first main result of his paper. The proof is quite general and relies heavily on Glivenko's Theorem, and the distinction between its status in minimal and intuitionistic logic may be taken as an explanation for the discrepant results that will be presented for both.

Theorem 4.4. $\Gamma \models^i_{STM} A$ *if and only if* $\Gamma \models_c A$.

Proof. The left-to-right direction follows immediately from the fact that the set of all classical interpretations is a subset of the set of all intuitionistic interpretations. $\Gamma \models^i_{STM} A$ implies that no intuitionistic interpretation makes all the formulas of Γ true and A false, which also implies that no classical interpretation makes all the formulas in Γ true and A false. Since classical interpretations collapse into usual truth-functional semantic, we immediatelly have $\Gamma \models_c A$.

Now for the right-to-left direction. Assume, for contradiction, that for some Γ and A we have $\Gamma \models_c A$ and $\Gamma \not\models^i_{ST} A$. By Lemma 4.1 we have $\Gamma \models_i \neg\neg A$, and so every intuitionistic interpretation which makes the formulas in Γ true also make $\neg\neg A$ true. Since $\Gamma \not\models^i_{ST} A$, by Definitions 3.13 and 3.16, there must be an intuitionistic interpretation I which makes all formulas in Γ true but makes A false. By Definition 3.9, A is false in a intuitionistic interpretation if and only if $\neg A$ is true. Then, both $\neg A$ and $\neg\neg A$ are true in I and thus receive value 1 in all its worlds, and so $\bot$ must also receive value 1 in all w due to the semantic clause for implication. But $\bot$ cannot receive the value 1 in intuitionistic interpretations. Contradiction. Thus, if $\Gamma \vDash_c A$ then $\Gamma \vDash^i_{ST} A$. □

The structure of this proof also makes it evident that it is not restricted to the characterization of intuitionistic logic via Kripke models, as no feature of those models is used in an essential fashion. Since it relies only on Glivenko's Theorem and the definition of falsity of a formula in an interpretation, every semantic for intuitionistic logic in which falsity of A is defined as equivalent to truth of $\neg A$ will experience this collapse. The particular relevance of this is made evident when one points out that some logicians denounce the inadequacy of Kripke models for formalizing intuitionistic semantics [39], which renders a proof which does not depend on the inner workings of such models especially interesting. It is completely inessential that our result uses a intuitionistic definition of the notion of "truth" formulated in Kripke models, as any notion of "construction" in which falsity/refutability of A were to be equated with a "construction" of $\neg A$ and truth/provability of "A" were to be equated with a "construction" of A would face the same issue.

We claim this result provides a partial answer to the question raised by Fitting in [14, pg. 393] of whether there is such a thing as an intuitionistic strict-tolerant logic, as any combination of the strict-tolerant definitions (if tolerant truth is equated with non-falsity) with semantics for intuitionistic logic which equate truth of $\neg A$ with falsity of A will collapse into classical logic at the inferential level. The negative results also apply to Fitting's proposal [14, pg. 393] of a strict-tolerant logic in which the conclusion of an inference is required to be classically true whenever the premises are intuitionistically true: classical truth of A is equivalent to classical truth of $\neg\neg A$, which in turn is equivalent to the non-falsity of A, leading us back to the same strict-tolerant definition of inference and thus to the same issues. On the other hand, Fitting himself has shown that such results can effectively be sidesteped through more nuanced definitions [15], which is why our results are partial.

To conclude commentaries on our mains results regarding intuitionistic logic, we point out that Glivenko's theorem can also be combined with intuitionistic proofs of equivalence between $\neg A$ and $\neg\neg\neg A$ to obtain his second theorem, $\vDash_c \neg A \Leftrightarrow \vDash_i \neg A$, which we may also generalize using the procedure of Lemma 4.1 to obtain $\Gamma \vDash_c \neg A \Leftrightarrow \Gamma \vDash_i \neg A$. These results bind both negations very closely together and present many challenges to their technical (and perhaps even conceptual) separation. In fact, since in many cases they are prone to collapse [7], many mixed systems – such as those used in ecumenical approaches to logic [32][22][5] – do not distinguish between classical and intuitionistic negation at all. It is evident, then, that strict-tolerant definitions can only be used together with intuitionistic logic without producing collapses if one either does not use non-falsity when evaluating the consequent of strict-tolerant inferences or promote a significant overhaul of the intuitionistic concepts of falsity or negation.

We now proceed to prove results concerning minimal logic.

Theorem 4.5. $\Gamma \vDash^m_{ST} A$ *implies* $\Gamma \vDash_c A$.

Proof. Trivial. We use the same reasoning as in the left-to-right direction of Theorem 4.4, noting that both the set of all intuitionistic interpretations and the set of all classical interpretations are subsets of the set of all minimal interpretations. □

Theorem 4.6. *There is a* Γ *and an* A *such that* $\Gamma \vDash_c A$ *but* $\Gamma \nvDash^m_{STM} A$.

Proof. Let Γ be the empty set and let formula A be the classical tautology $\neg A \to (A \to B)$. Consider now the minimal interpretation used in the proof of Lemma 4.2 and Corollary 4.3. According to Definition 3.9, $\neg(\neg a \to (a \to b))$ is true in this interpretation, as we have both $v_w(\neg(\neg a \to (a \to b))) = 1$ and $v_{w'}(\neg(\neg a \to (a \to b))) = 1$. Likewise, Definition 3.9 makes $\neg a \to (a \to b)$ false in this interpretation, and so $\nvDash^m_{ST} \neg a \to (a \to b)$. □

It is particularly interesting to notice that, in many traditional sources (such as Heyting's original monograph [19],) $\neg A \to (A \to B))$ is precisely the axiom taken from intuitionistic logic in order to obtain minimal logic.

In light of those basic results, one would naturally be led to ask what kind of logic is characterized by the combination of strict-tolerant inferences with minimal logic. Is it a paraconsistent logic? Does it have partial classical behavior? How much intuitionistic behavior it retains?

The answer, it turns out, is very surprising:

Theorem 4.7. *For any* Γ *and* A, $\Gamma \nvDash^m_{ST} A$.

Proof. Consider the interpretation with a single world w in which every atom is true, including $\bot$. Definition 3.8 can then be used to prove inductively that every molecular formula is also true. But, according to Definition 3.9, this means that all formulas in Γ are true and $\neg A$ is true, and so A is false. Thus, $\Gamma \nvDash^m_{ST} A$. □

Even though Theorem 4.6 was proved by considering the structure of a concrete minimal interpretation, it is actually a vacuous result. As such, the logic obtained by applying the strict-tolerant method to minimal logic has no valid inferences, much like the logic TS analyzed in [1].

Fortunately, these results do not carry over to the metainferential level, in which we indeed have valid consequences. Consider, for example, the following metainferential rule, in which the inference above the line is taken to be the antecedent of Definition 3.18 (that is, the sequence of inferences to the left of $\Rightarrow^*$) and the inference below the line is taken to be its succedent (that is, the inference to the right of $\Rightarrow^*$):

$$\frac{\Rightarrow A}{\Rightarrow A}$$

The interpretation considered in Theorem 4.7, Lemma 4.2 and Corollary 4.3 cannot be used to invalidate metainferences containing inferences with a non-empty set of premises: since it makes every formula both true and false, it invalidates any such inferences on the antecedent of the metainference, thus making the metainference itself hold vacuously in this interpretation by satisfying clause (i) of Definition 3.18. For all other interpretations it is evident that, conditional on A not being false, A is not false, and the results follows from the mere existence of such an interpretation (consider, for example, the interpretation containing a single world w with $v_w(A) = 1$ but $v_w(\bot) = 0$)).

In the context of the identity criteria proposed in [3], even though the logics obtained from classical and intuitionistic logic by the strict-tolerant method cannot be distinguished at the inferential level, this level is already sufficient to make a distinction between them and miminal logic. From the distinction at the inferential level we also immediately obtain distinction at the metainferential level, as validity of metainferences having an empty set of premises (that is, containing an empty sequence of inferences on its antecedent) is reducible to validity of the inference on the succedent, as briefly noted in the commentary presented immediatelly after Definition 3.18. However, we still need to investigate what happens in the metainferential level of intuitionistic logic, since it could be the case that it cannot be differentiated from classical logic.

Theorem 4.8. *For some Θ and some S we have $\Theta \vDash^c_{STM} S$ but $\Theta \nvDash^i_{STM} S$ and $\Theta \nvDash^m_{STM} S$.*

Proof. Consider the following metainference, in which the inferences of Θ stand above the line and S stands below:

$$\frac{\Rightarrow A \qquad \Rightarrow B}{\Rightarrow A \wedge B}$$

It is easy to verify that this metainference is valid in the case of classical logic, as any classical interpretation making the conclusion of both inferences occurring above the line non-false makes both A and B true and thus also $A \wedge B$ true (hence also non-false). In intuitionistic and minimal logic, however, it is not valid – which, as will be shown, essentially follows from the partial failure of DeMorgan's Laws, as in both logics we have $\neg(A \wedge B) \not\models \neg A \vee \neg B$.

Let I be a minimal interpretation with three worlds (w, w' and w''). For this interpretation, consider the following specifications, with a and b atomic:

(I) Only wRw', wRw'' and the relations obtained through reflexivity and transitivity of R hold;

(II) $v_w(a) = v_w(b) = v_w(\bot) = 0$;

(III) $v_{w'}(a) = 1$ and $v_{w'}(b) = v_{w'}(\bot) = 0$;

(IV) $v_{w''}(a) = v_{w''}(\bot) = 0$ and $v_{w''}(b) = 1$;

We can use the semantic clauses to show that neither a nor b are false in this interpretation, as $v_{w'}(\neg a) = 0$, $v_{w''}(\neg b) = 0$ and falsity requires assignment of 1 to the negation in all worlds. As such, this interpretation validates the semantic inferences $\Rightarrow a$ and $\Rightarrow b$. However, in no world of this interpretation we have both $v(a) = 1$ and $v(b) = 1$, which makes $a \wedge b$ false in the interpretation. Thus, since this atomic instance of the metainference is invalid, this metainference in general is invalid in minimal logic. Furthermore, since this interpretation is both a minimal and a intuitionistic interpretation, it is also invalid in intuitionistic logic. □

To conclude this section, it is worth noticing that the logics defined at the metainferential level by intuitionistic and minimal logic have a particular paraconsistent flavour, allowing us to draw comparisons similar to those in [1] concerning validation and invalidation of distinct formulations of the principle of explosion at the inferential and metainferential level.

From Theorems 4.4 and 4.7 it follows that $A, \neg A \Rightarrow B$ is valid on the system obtained from intuitionistic logic but not on the one obtained from minimal logic, even though they seem to have similar metainferential behaviour. Moreover, consider the following two metainferences:

$$\frac{\Rightarrow \bot}{\Rightarrow B} \qquad\qquad \frac{\Rightarrow A \qquad \Rightarrow \neg A}{\Rightarrow B}$$

Both logics validate the left one, but not the right one.

Since $\neg\bot$ is defined as $\bot \to \bot$, it will receive value 1 at all worlds w regardless of the value of $\bot$, and so $\bot$ is false in all minimal and intuistionitic interpretations –

which makes the inference on the left hold vacuously. However, for a counterexample to the second metainference, consider an interpretation with three worlds w, w' and w'', with wRw' and wRw'' but neither $w'Rw''$ nor $w''Rw'$ and in which $\bot$ and b receives value 0 at all worlds and a receives value 1 only at w'. Since $v_{w'}(\neg a) = 0$, a is not false, and since $v_{w''}(\neg a) = 1$ and $v_{w''}(\bot) = 0$ we have $v_{w''}(\neg\neg a) = 0$, and thus $\neg a$ is also not false. Since b receives value 0 at all worlds, $\neg b$ also receives value 1, and so b is false.

Since Theorem 4.8 also establishes a meaningful difference between those logics and the ones defined by the metainferences of classical logic, it follows that those are indeed some brand new logics with paraconsistent features. The proof of Theorem 4.8 also hints that the constructive nature of the underlying definitions has concrete effects on the notion of metainferential validity, as the particular metainference we considered seems to fail essentially due to the intuitionistic/minimal failure of one of the directions of the DeMorgan equivalences.

5 Conclusion

The results presented in this paper show that some particular features of intuitionistic and minimal systems interact with strict-tolerant definitions in unexpected ways. Glivenko's theorems and the proximity between classical and intuitionistic negation makes it so that intuitionistic strict-tolerant inferences easily collapse into classical inferences, and minimal logic's feature of allowing trivial models able to validate any premise and invalidate any conclusion makes it so that every minimal strict-tolerant inference is invalid. However, those features are partially eliminated when we go from the inferential to the metainferential level. Even though some behaviours observed at the inferential level are carried over to the metainferential one, metainferences provide enough wiggle room for those systems to escape their respective pathologies. As such, we are still able to obtain interesting logics from those combinations.

Our use of Glivenko's theorems in the results for intuitionistic logic makes them highly general, as every semantic definition in which falsity of A is equated with truth of $\neg A$ will face similar issues. This is not an absolute result, as shown by the success of Fitting's approach [15]. Nevertheless, the failure of DeMorgan's laws in intuitionistic and minimal logic also impacts what our logics validate at the metainferential level, preventing the total collapse into either classical or a trivial logic.

Intuitively, the results for intuitionistic logic show that, in light of the equivalence between truth of classical negation and truth of intuitionistic negation, the further equivalence between falsity and truth of negation makes logics defined through recourse to intuitionistic or classical falsity prone to a specific kind of collapse. Since classical truth of A is also equivalent to classical falsity of $\neg A$, the definition does not even have to reference falsity explicitly, which explains why the collapse happens both in strict-tolerant intuitionistic logic and in logics applying intuitionistic truth standards to the premises of an inference and classical truth standard to its conclusion. Moreover, the results for minimal logic show that logics in which truth and falsity may simultaneously obtain are prone to a different kind of collapse when we use them to define logics that essentially rely on distinctions between the two truth values.

In short, our work provides both a positive and a negative result to the literature, showing that the combination of strict-tolerant definitions with semantics for constructive logics are problematic but not entirely so. It also shows that strict-tolerant criterias of inferential validity cannot be taken to justify intuitionistic or minimal logic. Furthermore, the general nature of the proofs we present provide a cautionary tale: one should be very careful of mixing the strict-tolerant approach with logics that allow trivial models or which have a negation similar to that of classical logic, as those combinations are particularly prone to collapses.

References

[1] Eduardo Barrio, Federico Pailos, and Damian Szmuc. What is a paraconsistent logic?: Studia logica library. In Walter Carnielli and Malinowski Jar, editors, *Between consistency and inconsistency*, Trends in Logic, pages 89–108. Springer, Dordrecht, 01 2018.

[2] Eduardo Barrio, Lucas Rosenblatt, and Diego Tajer. The logics of strict-tolerant logic. *Journal of Philosophical Logic*, 44(5):551–571, 2015.

[3] Eduardo Alejandro Barrio, Federico Pailos, and Damian Szmuc. A hierarchy of classical and paraconsistent logics. *Journal of Philosophical Logic*, 49(1):93–120, 2020.

[4] Eduardo Alejandro Barrio, Federico Pailos, and Damian Szmuc. A recovery operator for nontransitive approaches. *The Review of Symbolic Logic*, 13(1):80–104, 2020.

[5] Victor Barroso-Nascimento, Luiz Carlos Pereira, and Elaine Pimentel. An ecumenical view of proof-theoretic semantics. *Synthese*, 206(4):197, Oct 2025.

[6] Victor Barroso-Nascimento, Ekaterina Piotrovskaya, and Elaine Pimentel. A proof-theoretic approach to the semantics of classical linear logic, 2025.

[7] David Binder and Thomas Piecha. Popper's notion of duality and his theory of negations. *History and Philosophy of Logic*, 38(2):154–189, 2017.

[8] Robert Brandom. Inferentialism and some of its challenges. *Philosophy and Phenomenological Research*, 74(3):651–676, 2007.

[9] Pablo Cobreros, Paul Egré, David Ripley, and Robert van Rooij. Tolerant, classical, strict. *Journal of Philosophical Logic*, 41(2):347–385, 2012.

[10] Michael Dummett. The philosophical basis of intuitionistic logic. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium '73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 5–40. Elsevier, Amsterdam, 1975.

[11] Michael Dummett. *The Logical Basis of Metaphysics.* William James Lectures. Harvard University Press, Cambridge, 1991.

[12] Michael Dummett. Truth from the constructive standpoint. *Theoria*, 64(2-3):122–138, 1998.

[13] Rodolfo C. Ertola and Marta Sagastume. Subminimal logic and weak algebras. *Reports Math. Log.*, 44:153–166, 2009.

[14] Melvin Fitting. A family of strict/tolerant logics. *Journal of Philosophical Logic*, 50(2):363–394, 2021.

[15] Melvin Fitting. Simple tableaus for simple intuitionistic logics. *Notre Dame Journal of Formal Logic*, Forthcoming.

[16] Alexander V. Gheorghiu. Proof-theoretic semantics for first-order logic, *Logic Journal of the IGPL* 33(5), 2025.

[17] Alexander V. Gheorghiu, Tao Gu, and David J. Pym. Proof-theoretic semantics for intuitionistic multiplicative linear logic. In Revantha Ramanayake and Josef Urban, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 367–385, Cham, 2023. Springer Nature Switzerland.

[18] Valery Glivenko. Sur quelques points de la logique de M. Brouwer. *Bulletin de la classe des sciences*, 5(25):183–188, 1929.

[19] Arend Heyting. *Intuitionism: an Introduction.* North-Holland, Amsterdam, 1956.

[20] Ulf Hlobil. A truth-maker semantics for st: Refusing to climb the strict/tolerant hierarchy. *Synthese*, 200(5):1–23, 2022.

[21] Ingebrigt Johansson. Der minimalkalkül, ein reduzierter intuitionistischer formalismus. *Compositio Mathematica*, 4:119–136, 1937.

[22] Sonia Marin, Luiz Carlos Pereira, Elaine Pimentel, and Emerson Sales. Ecumenical modal logic, 2020.

[23] Per Martin-Löf. Truth of a proposition, evidence of a judgement, validity of a proof. *Synthese*, 73(3):407–420, 1987.

[24] Julien Murzi and L. Shapiro. Validity and truth-preservation. In Theodora Achourioti, Henri Galinon, J. Martinez Fernandez, and K. Fujimoto, editors, *Unifying the Philosophy of Truth.* Springer, Dordrecht, The Netherlands, July 2015.

[25] Sergei Odintsov and Vladimir Rybakov. Unification and admissible rules for paraconsistent minimal Johanssons' logic J and positive intuitionistic logic IPC^+. *Annals of Pure and Applied Logic*, 164:771–784, 08 2013.

[26] Hiroakira Ono. Glivenko theorems revisited. *Annals of Pure and Applied Logic*, 161(2):246–250, 2009. Festschrift on the occasion of Franco Montagna's 60th birthday.

[27] Tomasz Placek. *Dummett's Case for Intuitionism*, pages 147–193. Springer Netherlands, Dordrecht, 1999.

[28] Dag Prawitz. Meaning and proofs: on the conflict between classical and intuitionistic logic. *Theoria*, 43(1):2–40, 1977.

[29] Dag Prawitz. Logical consequence from a constructivist view. In *The Oxford Handbook of Philosophy of Mathematics and Logic*. Oxford University Press, Oxford, 06 2007.

[30] Dag Prawitz. *Truth and Proof in Intuitionism*, pages 45–67. Springer Netherlands, Dordrecht, 2012.

[31] Dag Prawitz. *Validity of Inferences*, pages 179–204. De Gruyter, Berlin, Boston, 2013.

[32] Dag Prawitz. Classical versus intuitionistic logic. In Edward Haeusler, Wagner Sanz, and Bruno Lopes, editors, *Why is this a Proof?: Festschrift for Luiz Carlos Pereira*, number 27 in Tributes, pages 15–32. 2015.

[33] Dag Prawitz. The fundamental problem of general proof theory. *Studia Logica*, 107(1):11–29, 2019.

[34] Graham Priest. The logic of paradox. *Journal of Philosophical Logic*, 8(1):219–241, 1979.

[35] Graham Priest. *An Introduction to Non-Classical Logic: From If to Is*. Cambridge Introductions to Philosophy. Cambridge University Press, New York, 2nd edition, 2008.

[36] Stephen Read. Logical consequence as truth-preservation. *Logique et Analyse*, 46(183/184):479–493, 2003.

[37] Tor Sandqvist. Classical logic without bivalence. *Analysis*, 69(2):211–218, 04 2009.

[38] Tor Sandqvist. Base-extension semantics for intuitionistic sentential logic. *Logic Journal of the IGPL*, 23(5):719–731, 07 2015.

[39] Wagner Sanz. A criticism of Kripke's semantic for intuitionistic logic. *O Que Nos Faz Pensar*, 25:99, 01 2016.

[40] Peter Schroeder-Heister. Proof-Theoretic Semantics. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Stanford, Summer 2024 edition, 2024.

[41] Jonathan P. Seldin. Normalization and excluded middle. *Studia Logica: An International Journal for Symbolic Logic*, 48(2):193–217, 1989.

[42] Will Stafford and Victor Nascimento. Following all the rules: Intuitionistic completeness for generalized proof-theoretic validity. *Analysis*, 83(3):507–516, 07 2023.

[43] Tim van der Molen. The Johansson/Heyting letters and the birth of minimal logic. Technical Report X-2016-04, 2016.

Received 19 June 2024

Monoid Theory in Alonzo: A Little Theories Formalization in Simple Type Theory

William M. Farmer
McMaster University
wmfarmer@mcmaster.ca

Dennis Y. Zvigelsky
McMaster University
yankovsd@mcmaster.ca

Abstract

Alonzo is a practice-oriented classical higher-order version of predicate logic that extends first-order logic and that admits undefined expressions. Named in honor of Alonzo Church, Alonzo is based on Church's type theory, Church's formulation of simple type theory. The *little theories method* is a method for formalizing mathematical knowledge as a *theory graph* consisting of *theories* as nodes and *theory morphisms* as directed edges. The development of a mathematical topic is done in the "little theory" in the theory graph that has the most convenient level of abstraction and the most convenient vocabulary, and then the definitions and theorems produced in the development are transported, as needed, to other theories via the theory morphisms in the theory graph.

The purpose of this paper is to illustrate how a body of mathematical knowledge can be formalized in Alonzo using the little theories method. This is done by formalizing *monoid theory* — the body of mathematical knowledge about monoids — in Alonzo. Instead of using the *standard approach to formal mathematics* in which mathematics is done with the help of a proof assistant and all details are formally proved and mechanically checked, we employ an *alternative approach* in which everything is done within a formal logic but proofs are not required to be fully formal. The standard approach focuses on *certification*, while this alternative approach focuses on *communication* and *accessibility*.

We would like to thank the referee for reviewing this paper and providing very helpful suggestions.

1 Introduction

Formal mathematics is mathematics done within a formal logic. *Formalization* is the act of expressing mathematical knowledge in a formal logic. One of the chief benefits of formal mathematics is that a body of mathematical knowledge can be formalized as a precise, rigorous, and highly organized structure. This structure records the logical relationships between mathematical concepts and facts, how these concepts and facts are expressed in a given theory, and how one theory is related to another. Since it is based on a formal logic, it can be developed and analyzed using software.

An attractive and powerful method for organizing mathematical knowledge is the *little theories method* [22]. A body of mathematical knowledge is represented in the form of a *theory graph* [38] consisting of *theories* as nodes and *theory morphisms* as directed edges. Each mathematical topic is developed in the "little theory" in the theory graph that has the most convenient level of abstraction and the most convenient vocabulary. Then the definitions and theorems produced in the development are transported, as needed, from this abstract theory to other, usually more concrete, theories in the graph via the theory morphisms in the graph.

The *standard approach to formal mathematics* focuses on *certification*: Mathematics is done with the help of a proof assistant and all details are formally proved and mechanically checked. We present in Section 2 an *alternative approach to formal mathematics*, first introduced in [21], that focuses on two other goals: *communication* and *accessibility*. The idea is that everything is done within a formal logic but proofs are not required to be fully formal and the entire development is optimized for communication and accessibility. We believe that formal mathematics can be made more useful, accessible, and natural to a wider range of mathematics practitioners — mathematicians, computing professionals, engineers, and scientists who use mathematics in their work — by pursuing this alternative approach.

The purpose of this paper is to illustrate how a body of mathematical knowledge can be formalized in Alonzo [20], a practice-oriented classical higher-order logic that extends first-order logic, using the little theories method and the alternative approach to formal mathematics. Named in honor of Alonzo Church, Alonzo is based on Church's type theory [8], Church's formulation of simple type theory [18], and is closely related to Peter Andrews' $\mathcal{Q}_0$ [1]; $\mathcal{Q}_0^{\mathrm{u}}$ [17], a version of $\mathcal{Q}_0$ with undefined expressions; and LUTINS [13, 14, 15], the logic of the IMPS proof assistant [23, 24]. Unlike traditional predicate logics, Alonzo admits partial functions and undefined expressions in accordance with the approach employed in mathematical practice that we call the *traditional approach to undefinedness* [16]. Since partial functions naturally arise from theory morphisms [15], the little theories method works best with a logic like Alonzo that supports partial functions.

Alonzo has a simple syntax with a *formal notation* for machines and a *compact notation* for humans that closely resembles the notation found in mathematical practice. The compact notation is defined by the extensive set of *notational definitions and conventions* given in [20]. Alonzo has two semantics, one for mathematics based on *standard models* and one for logic based on Henkin-style *general models* [32]. By virtue of its syntax and semantics, Alonzo is exceptionally well suited for expressing and reasoning about mathematical ideas and for specifying mathematical structures. A brief overview of the syntax and semantics of Alonzo is given in Section 3. See [20] for a full presentation of Alonzo.

We have chosen *monoid theory* — the concepts, properties, and facts about monoids — as a sample body of mathematical knowledge to formalize in Alonzo. A *monoid* is a mathematical structure consisting of a nonempty set, an associative binary function on the set, and a member of the set that is an identity element with respect to the function. Monoids are abundant in mathematics and computing. Single-object categories are monoids. Groups are monoids in which every element has an inverse. And several algebraic structures, such as rings, fields, Boolean algebras, and vector spaces, contain monoids as substructures.

Since a monoid is a significantly simpler algebraic structure than a group, monoid theory lacks the rich structure of group theory. We are formalizing monoid theory in Alonzo, instead of group theory, since it has just enough structure to adequately illustrate how a body of mathematical knowledge can be formalized in Alonzo. We will see that employing the little theories method in the formalization of monoid theory in Alonzo naturally leads to a robust theory graph.

Alonzo is equipped with a set of *mathematical knowledge modules* (*modules* for short) for constructing various kinds of mathematical knowledge units. For example, it has modules for constructing "theories" and "theory morphisms". A *language* (or *signature*) of Alonzo is a pair $L = (\mathcal{B}, \mathcal{C})$, where $\mathcal{B}$ is a finite set of base types and $\mathcal{C}$ is a set of constants, that specifies a set of expressions. A *theory* of Alonzo is a pair $T = (L, \Gamma)$ where L is a language called the *language of* T and Γ is a set of sentences of L called the *axioms of* T. And a *theory morphism* of Alonzo from a theory T_1 to a theory T_2 is a mapping of the expressions of T_1 to the expressions of T_2 such that (1) base types are mapped to types and closed quasitypes (expressions that denote sets of values), (2) constants are mapped to closed expressions of appropriate type, and (3) valid sentences are mapped to valid sentences.

Alonzo also has modules for constructing "developments" and "development morphisms". A *theory development* (or *development* for short) of Alonzo is a pair $D = (T, \Xi)$ where T is a theory and Ξ is a (possibly empty) sequence of definitions and theorems presented, respectively, as definition and theorem packages (see [20, Section 12.1]). T is called the *bottom theory* of D, and T', the extension of T obtained by adding the definitions in Ξ to T, is called the *top theory* of D. We say that D is a *development of* T. A *development morphism* from a development D_1 to a development D_2 is a partial mapping from the expressions of D_1 to the expressions of D_2 that restricts to a theory morphism from the bottom theory of D_1 to the bottom theory of D_2 and that canonically extends to a theory morphism from the top theory of D_1 to the top theory of D_2 (see [20, Section 14.4.1]). Theories and theory morphisms are special cases of developments and development morphisms, respectively, since we identify a theory T with the trivial development $(T, [\,])$.

The modules for constructing developments and development morphisms provide the means to represent knowledge in the form of a *development graph*, a richer kind of theory graph, in which the nodes are developments and the directed edges are development morphisms. Alonzo includes modules for transporting definitions and theorems from one development to another via development morphisms. The design of Alonzo's module system is inspired by the IMPS implementation of the little theories method [22, 23, 24].

The formalization of monoid theory presented in this paper exemplifies an *alternative approach to formal mathematics.* We validate the definitions and theorems in a development using traditional (nonformal) mathematical proof. However, we extensively use the axioms, rules of inference, and metatheorems of $\mathfrak{A}$ — the formal proof system for Alonzo presented in [20] which is derived from Andrews' proof system for $\mathcal{Q}_0$ [1] — in these traditional proofs. The proofs are not included in the modules used to construct developments. Instead, they are given separately in Appendix A.

We produced the formalization of monoid theory with just a minimal amount of software support. We used the set of LaTeX macros and environments for Alonzo given in [19] plus a few macros created specifically for this paper. The macros are for presenting Alonzo types and expressions in both the formal and compact notations. The environments are for presenting Alonzo mathematical knowledge modules. The Alonzo modules are printed in brown color.

The overarching goal of this paper is to demonstrate that, using the little theories method and the alternative approach to formal mathematics, mathematical knowledge can be very effectively formalized in a version of simple type theory like Alonzo. Specifically, we want to show the following:

1. How the little theories method can be used to organize mathematical knowledge so that clarity is maximized and redundancy is minimized.

2. How formal libraries of mathematical knowledge that prioritize communication over certification can be built using the alternative approach to formal

mathematics with tools that are much simpler to learn and use than a proof assistant.

3. How Alonzo is exceptionally well suited for expressing and reasoning about mathematical ideas and for specifying mathematical structures in a direct and natural manner.

The paper is organized as follows. We present in Section 2 the alternative approach to formal mathematics and argue that this kind of approach can better serve the average mathematics practitioner than the standard approach. Section 3 gives a brief presentation of the syntax and semantics of Alonzo. Sections 4–11 present developments of theories of monoids, commutative monoids, transformation monoids, monoid actions, monoid homomorphisms, and monoids over real number arithmetic plus some supporting developments. These developments have been constructed to be illustrative; they are not intended to be complete in any sense. Sections 4–11 also present various development morphisms that are used to transport definitions and theorems from one development to another. Section 12 shows how our formalization of monoid theory can support a theory of strings. Related work is discussed in Section 13. The paper concludes in Section 14 with a summary and some final remarks. The definitions and theorems of the developments we have constructed are validated by traditional mathematical proofs presented in Appendix A. Appendix B contains some miscellaneous theorems needed for the proofs in Appendix A.

2 Alternative approach to formal mathematics

A *formal logic* (*logic* for short) is a *family of languages* such that:

1. The languages of the logic have a *common precise syntax*.

2. The languages of the logic have a *common precise semantics with a notion of logical consequence*.

3. There is a *sound formal proof system* for the logic in which proofs can be syntactically constructed.

Examples of formal logics for mathematics are the various versions of first-order logic, set theory, simple type theory, and dependent type theory.

There are five big benefits of formal mathematics, i.e., doing mathematics within a formal logic.

First, *mathematics can be done with greater rigor.* All mathematical ideas are expressed and reasoned about in a theory T of a formal logic. Mathematical concepts and statements are expressed as expressions and sentences of the language of T. All of these expressions and sentences have a precise, unambiguous meaning. The assumptions underlying the reasoning about the mathematical ideas are made explicit as axioms of the theory. The theorems of theory are precisely defined as the logical consequences of the axioms of the theory. And, finally, the theory is constructed so that it contains only the vocabulary and assumptions that are needed for the task at hand; irrelevant details are abstracted away.

Second, *conceptual errors can be systematically discovered.* In formal mathematics, all concepts and statements must be expressed in a language of a formal logic that has a precise semantics. The process of expressing mathematical ideas in a formal logic naturally leads to many conceptual errors being caught similarly to how type errors are caught in a modern programming language by type checking. Thus conceptual errors can be discovered systematically in formal mathematics in a way that is largely not possible in traditional mathematics. As a result, formal mathematics often yields a deeper understanding of the mathematics being explored than traditional mathematics.

Third, *mathematics can be done with software support.* Since the languages of a formal logic have a precise common syntax, the expressions and sentences of a language can be represented as data structures. The expressions and sentences can then be analyzed, manipulated, and processed via their representations as data structures. This, in turn, enables the study, discovery, communication, and certification of mathematics to be done with the aid of software. Since the languages also have a precise common semantics, there is a precise basis for verifying the correctness of this software.

Fourth, *results can be mechanically checked.* Formal proofs can be represented as data structures, and software can be used to check that one of these data structures represents an actual proof in the formal proof system of the logic. Software can also be used to help construct the formal proofs. Since the software needed to check the correctness of the formal proofs is often very simple and easily verified itself, it is possible to verify the correctness of the formal proofs with a very high level of assurance.

Fifth, *we can regard mathematical knowledge as a formal structure consisting of a network of interconnected theories.* A library of mathematical knowledge that represents this formal structure can be built by creating theories, defining new concepts, stating and proving theorems, and connecting one theory to another with theory morphisms that map the theorems of one theory to the theorems of another theory. The knowledge embodied in a structured library of this kind can be studied, managed, searched, and presented using software.

The benefits of formal mathematics are huge. Greater rigor and discovering conceptual errors have been principal goals of mathematicians for thousands of years. Software support can greatly extend the reach and productivity of mathematics practitioners. Mechanically checked results can drive mathematics forward in areas where the ideas are poorly understood (often due to their novelty) or highly complex. And mathematical knowledge as a formal structure can enable the techniques and tools of mathematics and computing to be applied to mathematical knowledge itself.

The standard approach to formal mathematics, in which mathematics is done with the help of a proof assistant and all details are formally proved and mechanically checked, has three major strengths:

1. It achieves all five benefits of formal mathematics mentioned above.
2. All theorems are verified by machine-checked formal proofs. Thus there is a very high level of assurance that the results produced are correct.
3. There are several powerful proof assistants available, such as HOL [29], HOL Light [31], , Isabelle/HOL [48], Lean [10], Metamath/ZFC [39], Mizar [42], and Rocq (formerly Coq) [54], that support the approach.

It also has two important weaknesses:

1. It prioritizes certification over communication. For the average mathematics practitioner, communicating mathematical ideas is usually much more important than certifying mathematical results when the mathematics is well understood.
2. It is not accessible to the great majority of mathematics practitioners. Having to learn a strange logic and work with a complex proof assistant that utilizes unfamiliar ways of expressing and reasoning about mathematics is very often a bridge too far for the average mathematics practitioner.

We strongly believe, as an alternative to the standard approach, an approach to formal mathematics is needed that focuses on two goals, communication and accessibility, the weaknesses of the standard approach. To achieve these goals the alternative approach should satisfy the following requirements:

R1. *The underlying logic is fully formal and supports standard mathematical practice.* Supporting mathematical practice makes the logic easier to learn and use and makes formalization a more natural process.

R2. *Proofs can be traditional, formal, or a combination of the two.* This flexibility in how proofs are written enables proofs to be a vehicle for communication as well as certification.

R3. *There is support for organizing mathematical knowledge using the little theories method.* This enables mathematical knowledge to be formalized to maximize clarity and minimize redundancy.

R4. *There are several levels of supporting software.* The levels can range from just LaTeX support to a full proof assistant. The user can thus choose the level of software support they want to have and the level of investment in learning the software they want to make.

The alternative approach can achieve all five benefits of formal mathematics mentioned above, but it cannot achieve the same level of assurance as the standard approach that the results produced are correct. This is because the alternative approach prioritizes communication and accessibility over certification. Since most mathematics practitioners are usually more concerned about communication and accessibility than certification, the alternative approach is on average a better approach to formal mathematics than the standard approach. This is particularly true for applications that involve well-understood mathematics, the kind of mathematics that arises in mathematics education and routine applications. However, when the certification of results is the most important concern, the standard approach will often be a better choice than the alternative approach.

This paper employs an implementation of the alternative approach based on Alonzo that satisfies the first three requirements and partially satisfies the fourth requirement. Alonzo is a form of predicate logic, which is widely familiar to mathematics practitioners. Moreover, it supports the reasoning instruments that are most common in mathematical practice including functions, sets, tuples, and lists; mathematical structures; higher-order and restricted quantification; definite description; theories and theory morphisms; definitional and other kinds of conservative extensions; inductive sets; notational definitions and conventions, and undefined expressions. Thus Alonzo satisfies R1 as well or better than almost any other logic.

R2 is satisfied by our implementation of the alternative approach since proofs can be traditional or formal. Thus communication can be prioritized over certification in proofs when the mathematics is well understood. In this paper, all the proofs are traditional, but some make use of the axioms, rules of inference, and metatheorems of $\mathfrak{A}$, the proof system of Alonzo.

R3 is satisfied since Alonzo is equipped with a module system for organizing mathematical knowledge using the little theories method.

Our implementation of the alternative approach provides only the simplest level of software support: LaTeX macros for presenting Alonzo types and expressions and LaTeX environments for presenting Alonzo modules. Other levels of software support are possible; see the discussion in Chapter 16 of [20]. Alonzo has not been implemented in a proof assistant, but since it is closely related to LUTINS [13, 14, 15], the logic of the IMPS proof assistant [23, 24], it could be implemented in much the same way that LUTINS is implemented in IMPS. Thus R4 is only partially satisfied now, but it could be fully satisfied with the addition of more levels of software support.

The great majority of mathematics practitioners — including mathematicians — are much more interested in communicating mathematical ideas than in formally certifying mathematical results. Hence, the alternative approach — with support for standard mathematical practice, traditional proofs, the little theories method, and several levels of software — is likely to serve the needs of the average mathematics practitioner much better than the standard approach. This is especially true when the mathematical knowledge involved is well understood (such as monoid theory) and certification via traditional proof is adequate for the purpose at hand.

In summary, we believe that the alternative approach is not a replacement for the standard approach, but it would be more useful, accessible, and natural than the standard approach for the vast majority of mathematics practitioners.

3 Alonzo

Alonzo is fully presented in [20]. Due to space limitations, we cannot duplicate the entire presentation of Alonzo in this paper. Ideally, the reader should be familiar with the syntax and semantics of Alonzo presented in Chapters 4–7; the proof system for Alonzo presented in Chapter 8 and Appendices A–C; the tables of notational definitions found in Chapters 4, 6, 11, and 13; the notational conventions presented in Chapters 4 and 6; and the various kinds of (mathematical knowledge) modules of Alonzo presented in Chapters 9, 10, 12, and 14. However, we will give in this section a brief presentation of the syntax and semantics of Alonzo with most of the text taken from Chapters 4–6 of [20].

3.1 Syntax

The syntax of Alonzo consists of "types" that denote nonempty sets of values and "expressions" that either denote values (when they are defined) or denote nothing at all (when they are undefined). We present the syntax of Alonzo types and expressions with the compact notation, an "external" syntax intended for humans. The reader is referred to [20] for the formal syntax, an "internal" syntax intended for machines. The compact notation for types and expressions is given below. Additional compact notation is introduced using *notational definitions* and *notational conventions*. A *notational definition* has the form

A stands for B,

where A and B are notations that present types or expressions; it defines A to be an alternate — and usually more compact, convenient, or standard — notation for presenting the type or expression that B presents. The meaning of A is the meaning of B. The notational definitions are given in tables with boxes surrounding the definitions, and the notational conventions are assigned names of the form "Notational Convention n".

Let $\mathcal{S}_{\mathsf{bt}}$, $\mathcal{S}_{\mathsf{var}}$, $\mathcal{S}_{\mathsf{con}}$ be fixed countably infinite sets of symbols that will serve as names of base types, variables, and constants, respectively. We assume that $\mathcal{S}_{\mathsf{bt}}$ contains the symbols $A, B, C \ldots, X, Y, Z$, etc., $\mathcal{S}_{\mathsf{var}}$ contains the symbols $a, b, c \ldots, x, y, z$, etc., and $\mathcal{S}_{\mathsf{con}}$ contains the symbols $A, B, C \ldots, X, Y, Z$, etc., numeric symbols, nonalphanumeric symbols, and words in lowercase sans sarif font.[1] We will employ the following syntactic variables for these symbols as well as types and expressions which are defined just below:

1. $\mathbf{a}, \mathbf{b}$, etc. range over $\mathcal{S}_{\mathsf{bt}}$.
2. $\mathbf{f}, \mathbf{g}, \mathbf{h}, \mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{m}, \mathbf{n}, \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{x}, \mathbf{y}, \mathbf{z}$, etc. range over $\mathcal{S}_{\mathsf{var}}$.
3. $\mathbf{c}, \mathbf{d}$, etc. range over $\mathcal{S}_{\mathsf{con}}$.
4. $\alpha, \beta, \gamma, \delta$, etc. range over types.
5. $\mathbf{A}_\alpha, \mathbf{B}_\alpha, \mathbf{C}_\alpha, \ldots, \mathbf{X}_\alpha, \mathbf{Y}_\alpha, \mathbf{Z}_\alpha$, etc. range over expressions of type α.

A *type* of Alonzo is a string of symbols defined inductively by the following formation rules:

T1. *Type of truth values*: o is a type.

T2. *Base type*: $\mathbf{a}$ is a type.

T3. *Function type*: $(\alpha \rightarrow \beta)$ is a type.

T4. *Product type*: $(\alpha \times \beta)$ is a type.

[1] An expression like "u, v, w, etc." means the set of symbols that includes u, v, and w, and all possible annotated forms of u, v, and w such as u', v_1, and $\widetilde{w}$.

Let $\mathcal{T}$ denote the set of types of Alonzo. We assume $o \notin \mathcal{S}_{\mathsf{bt}}$.

When there is no loss of meaning, matching pairs of parentheses in the compact notation for types may be omitted (Notational Convention 1). We assume that function type formation associates to the right so that, e.g., a type of the form

$$(\alpha \to (\beta \to \gamma))$$

may be written more simply as $\alpha \to \beta \to \gamma$ (Notational Convention 2).

A type α denotes a nonempty set D_α of values. o denotes the set $D_o = \mathbb{B}$ of the Boolean (truth) values F and T. $(\alpha \to \beta)$ denotes some set $D_{\alpha\to\beta}$ of (partial and total) functions from D_α to D_β. $(\alpha \times \beta)$ denotes the Cartesian product $D_{\alpha\times\beta} = D_\alpha \times D_\beta$. We will use base types to denote the base domains of mathematical structures.

An *expression of type* α of Alonzo is a string of symbols defined inductively by the following formation rules:

E1. *Variable*: $(\mathbf{x} : \alpha)$ is an expression of type α.

E2. *Constant*: $\mathbf{c}_\alpha$ is an expression of type α.

E3. *Equality*: $(\mathbf{A}_\alpha = \mathbf{B}_\alpha)$ is an expression of type o.

E4. *Function application*: $(\mathbf{F}_{(\alpha\to\beta)}\, \mathbf{A}_\alpha)$ is an expression of type β.

E5. *Function abstraction*: $(\lambda\, \mathbf{x} : \alpha \,.\, \mathbf{B}_\beta)$ is an expression of type $(\alpha \to \beta)$.

E6. *Definite description*: $(\mathrm{I}\, \mathbf{x} : \alpha \,.\, \mathbf{A}_o)$ is an expression of type α where $\alpha \neq o$.

E7. *Ordered pair*: $(\mathbf{A}_\alpha, \mathbf{B}_\beta)$ is an expression of type $(\alpha \times \beta)$.

Let $\mathcal{E}$ denote the set of expressions of Alonzo. A *formula* is an expression of type o, and a *sentence* is a closed formula.

When there is no loss of meaning, matching pairs of parentheses in expressions may be omitted (Notational Convention 3). We assume that function application formation associates to the left so that, e.g., an expression of the form $((\mathbf{G}_{\alpha\to\beta\to\gamma}\, \mathbf{A}_\alpha)\, \mathbf{B}_\beta)$ may be written more simply as $\mathbf{G}_{\alpha\to\beta\to\gamma}\, \mathbf{A}_\alpha\, \mathbf{B}_\beta$ (Notational Convention 4). When the type α of a constant $\mathbf{c}_\alpha$ is known from the context of the constant, we will very often write the constant as simply $\mathbf{c}$ (Notational Convention 5). A variable $(\mathbf{x} : \alpha)$ occurring in the body $\mathbf{B}_\beta$ of $\lambda\, \mathbf{x} : \alpha \,.\, \mathbf{B}_\beta$ or in the body $\mathbf{A}_o$ of $\mathrm{I}\, \mathbf{x} : \alpha \,.\, \mathbf{A}_o$ may be written as just $\mathbf{x}$ if there is no resulting ambiguity (Notational Convention 6). So, for example, $\lambda\, \mathbf{x} : \alpha \,.\, (\mathbf{x} : \alpha)$ may be written more simply as $\lambda\, \mathbf{x} : \alpha \,.\, \mathbf{x}$. We will employ this convention for the other variable binders of Alonzo introduced later by notational definitions (Notational Convention 7). A variable $(\mathbf{x} : \alpha)$ occurring in $\mathbf{B}_\beta$ may be written as just $\mathbf{x}$ if the type α is known from the context of the occurrence of $(\mathbf{x} : \alpha)$ in $\mathbf{B}_\beta$ (Notational Convention 8). For example, $\mathbf{A}_\alpha = (\mathbf{x} : \alpha)$ may be written as $\mathbf{A}_\alpha = \mathbf{x}$.

An expression of type α is always defined if $\alpha = o$ and may be either defined or undefined if $\alpha \neq o$. If defined, it denotes a value in D_α, the denotation of α. If undefined, it denotes nothing at all. We will use constants to denote the distinguished values of mathematical structures.

As previously defined, a *language* (or *signature*) of Alonzo is a pair $L = (\mathcal{B}, \mathcal{C})$ where $\mathcal{B}$ is a finite set of base types and $\mathcal{C}$ is a set of constants $\mathbf{c}_\alpha$ where each base type occurring in α is a member of $\mathcal{B}$. A type α is a *type of* L if all the base types occurring in α are members of $\mathcal{B}$, and an expression $\mathbf{A}_\alpha$ is an *expression of* L if all the base types occurring in $\mathbf{A}_\alpha$ are members of $\mathcal{B}$ and all the constants occurring in $\mathbf{A}_\alpha$ are members of $\mathcal{C}$. Let $\mathcal{T}(L) \subseteq \mathcal{T}$ denote the set of types of L and $\mathcal{E}(L) \subseteq \mathcal{E}$ denote the set of expressions of L. Notice that $\mathcal{B}$ and $\mathcal{C}$ may be empty, but $\mathcal{T}(L)$ and $\mathcal{E}(L)$ are always nonempty since $o \in \mathcal{T}(L)$.

3.2 Semantics

Let $L = (\mathcal{B}, \mathcal{C})$ be a language of Alonzo. We will now define the semantics of L.

A *frame* for L is a collection $\mathcal{D} = \{D_\alpha \mid \alpha \in \mathcal{T}(L)\}$ of nonempty domains (sets) of values such that:

F1. *Domain of truth values*: $D_o = \mathbb{B} = \{\text{F}, \text{T}\}$.

F2. *Predicate domain*: $D_{\alpha \rightarrow o}$ is a set of *some* total functions from D_α to D_o for $\alpha \in \mathcal{T}(L)$.

F3. *Function domain*: $D_{\alpha \rightarrow \beta}$ is a set of *some* partial and total functions from D_α to D_β for $\alpha, \beta \in \mathcal{T}(L)$ with $\beta \neq o$.

F4. *Product domain*: $D_{\alpha \times \beta} = D_\alpha \times D_\beta$ for $\alpha, \beta \in \mathcal{T}(L)$.

A predicate domain $D_{\alpha \rightarrow o}$ is *full* if it is the set of *all* total functions from D_α to D_o, and a function domain $D_{\alpha \rightarrow \beta}$ with $\beta \neq o$ is *full* if it is the set of *all* partial and total functions from D_α to D_β. The frame is *full* if $D_{\alpha \rightarrow \beta}$ is full for all $\alpha, \beta \in \mathcal{T}(L)$. Notice that the only restriction on a *base domain*, i.e., $D_{\mathbf{a}}$ for some $\mathbf{a} \in \mathcal{B}$, is that it is nonempty and that the frame is completely determined by its base domains when the frame is full. An *interpretation* of L is a pair $M = (\mathcal{D}, I)$ where $\mathcal{D} = \{D_\alpha \mid \alpha \in \mathcal{T}(L)\}$ is a frame for L and I is an *interpretation function* that maps each constant in $\mathcal{C}$ of type α to an element of D_α. Notice that

$$(\{D_{\mathbf{a}} \mid \mathbf{a} \in \mathcal{B}\}, \{I(\mathbf{c}_\alpha) \mid \mathbf{c}_\alpha \in \mathcal{C}\})$$

is a mathematical structure. Hence an interpretation of a language *defines* (1) a mathematical structure and (2) a mapping of the base types and constants of the language to the base domains and distinguished values, respectively, of the mathematical structure.

Let $\mathcal{D} = \{D_\alpha \mid \alpha \in \mathcal{T}(L)\}$ be a frame for L. An *assignment into* $\mathcal{D}$ is a function φ whose domain is the set of variables of L such that $\varphi((\mathbf{x} : \alpha)) \in D_\alpha$ for each variable $(\mathbf{x} : \alpha)$ of L. Given an assignment φ, a variable $(\mathbf{x} : \alpha)$ of L, and $d \in D_\alpha$, let $\varphi[(\mathbf{x} : \alpha) \mapsto d]$ be the assignment ψ in $\mathcal{D}$ such that $\psi((\mathbf{x} : \alpha)) = d$ and $\psi((\mathbf{y} : \beta)) = \varphi((\mathbf{y} : \beta))$ for all variables $(\mathbf{y} : \beta)$ of L distinct from $(\mathbf{x} : \alpha)$. Given an interpretation M of L, let $\mathsf{assign}(M)$ be the set of assignments into the frame of M.

Let $\mathcal{D} = \{D_\alpha \mid \alpha \in \mathcal{T}(L)\}$ be a frame for L and $M = (\mathcal{D}, I)$ be an interpretation of L. M is a *general model* of L if there is a partial binary *valuation function* V^M such that, for all assignments $\varphi \in \mathsf{assign}(M)$ and expressions $\mathbf{C}_\gamma$ of L, (1) either $V^M_\varphi(\mathbf{C}_\gamma) \in D_\gamma$ or $V^M_\varphi(\mathbf{C}_\gamma)$ is undefined[2] and (2) each of the following conditions is satisfied:

V1. $V^M_\varphi((\mathbf{x} : \alpha)) = \varphi((\mathbf{x} : \alpha))$.

V2. $V^M_\varphi(\mathbf{c}_\alpha) = I(\mathbf{c}_\alpha)$.

V3. $V^M_\varphi(\mathbf{A}_\alpha = \mathbf{B}_\alpha) = \textsc{t}$ if $V^M_\varphi(\mathbf{A}_\alpha)$ is defined, $V^M_\varphi(\mathbf{B}_\alpha)$ is defined, and $V^M_\varphi(\mathbf{A}_\alpha) = V^M_\varphi(\mathbf{B}_\alpha)$. Otherwise, $V^M_\varphi(\mathbf{A}_\alpha = \mathbf{B}_\alpha) = \textsc{f}$.

V4. $V^M_\varphi(\mathbf{F}_{\alpha \rightarrow \beta}\, \mathbf{A}_\alpha) = V^M_\varphi(\mathbf{F}_{\alpha \rightarrow \beta})(V^M_\varphi(\mathbf{A}_\alpha))$ if $V^M_\varphi(\mathbf{F}_{\alpha \rightarrow \beta})$ is defined, $V^M_\varphi(\mathbf{A}_\alpha)$ is defined, and $V^M_\varphi(\mathbf{F}_{\alpha \rightarrow \beta})$ is defined at $V^M_\varphi(\mathbf{A}_\alpha)$. Otherwise, $V^M_\varphi(\mathbf{F}_{\alpha \rightarrow \beta}\, \mathbf{A}_\alpha) = \textsc{f}$ if $\beta = o$ and $V^M_\varphi(\mathbf{F}_{\alpha \rightarrow \beta}\, \mathbf{A}_\alpha)$ is undefined if $\beta \neq o$.

V5. $V^M_\varphi(\lambda\, \mathbf{x} : \alpha\ .\ \mathbf{B}_\beta)$ is the (partial or total) function $f \in D_{\alpha \rightarrow \beta}$ such that, for each $d \in D_\alpha$, $f(d) = V^M_{\varphi[(\mathbf{x}:\alpha) \mapsto d]}(\mathbf{B}_\beta)$ if $V^M_{\varphi[(\mathbf{x}:\alpha) \mapsto d]}(\mathbf{B}_\beta)$ is defined and $f(d)$ is undefined if $V^M_{\varphi[(\mathbf{x}:\alpha) \mapsto d]}(\mathbf{B}_\beta)$ is undefined.

V6. $V^M_\varphi(\mathrm{I}\, \mathbf{x} : \alpha\ .\ \mathbf{A}_o)$ is the $d \in D_\alpha$ such that $V^M_{\varphi[(\mathbf{x}:\alpha) \mapsto d]}(\mathbf{A}_o) = \textsc{t}$ if there is exactly one such d. Otherwise, $V^M_\varphi(\mathrm{I}\, \mathbf{x} : \alpha\ .\ \mathbf{A}_o)$ is undefined.

V7. $V^M_\varphi((\mathbf{A}_\alpha, \mathbf{B}_\beta)) = (V^M_\varphi(\mathbf{A}_\alpha), V^M_\varphi(\mathbf{B}_\beta))$ if $V^M_\varphi(\mathbf{A}_\alpha)$ and $V^M_\varphi(\mathbf{B}_\beta)$ are defined. Otherwise, $V^M_\varphi((\mathbf{A}_\alpha, \mathbf{B}_\beta))$ is undefined.

[2]We write $V^M_\varphi(\mathbf{C}_\gamma)$ instead of $V^M(\varphi, \mathbf{C}_\gamma)$.

V^M is unique when it exists. $V^M_\varphi(\mathbf{C}_\gamma)$ is called the *value of* $\mathbf{C}_\gamma$ *in* M *with respect to* φ when $V^M_\varphi(\mathbf{C}_\gamma)$ is defined. $\mathbf{C}_\gamma$ is said to have no value in M with respect to φ when $V^M_\varphi(\mathbf{C}_\gamma)$ is undefined.

An interpretation $M = (\mathcal{D}, I)$ of L is a *standard model* of L if $\mathcal{D}$ is full. Every standard model of L is a general model of L.

3.3 Additional compact notation

The compact notation for Alonzo types and expressions given above is extended in [20] with a variety of operators, binders, and abbreviations. Equipped with this additional compact notation, Alonzo becomes a practical logic in which mathematical ideas can be expressed naturally and succinctly. The compact notation that we need in this paper from Chapter 6 of [20] is presented in Tables 1–8. To make the notational definitions as readable as possible we have omitted matching parentheses in the right-hand side of the definitions when there is no loss of meaning and it is obvious where they should occur.

In Table 1, we present notation for the truth values and the standard Boolean operators. The notation $\wedge_{o \to o \to o}$ is an example of a *pseudoconstant*. It is not a real constant of Alonzo, but it stands for an expression $\mathbf{C}_\gamma$ that can be used just like a constant $\mathbf{c}_\gamma$. Unlike a normal constant, $\wedge_{o \to o \to o}$ and most other pseudoconstants can be employed in any language. Thus they serve as logical constants. The same symbols that are used to write constants are used to write pseudoconstants and parametric pseudoconstants (which are defined below) (Notational Convention 9).

In Table 2, we present notation for binary operators. We will occasionally use implicit notational definitions analogous to the notational definitions in Table 2 for the infix operators $<$, $>$, and $\geq$ corresponding to $\leq$ for other weak order operators such as $\subseteq$ and $\sqsubseteq$ (Notational Convention 10).

In Table 3, we present notation for the universal and existential quantifiers. We will usually write a sequence of universal quantifiers and a sequence of existential quantifiers in a more compact form with a single quantifier (Notational Convention 11). Thus, for example,

$$\forall\, \mathbf{x} : \alpha \,.\, \forall\, \mathbf{y} : \alpha \,.\, \forall\, \mathbf{z} : \beta \,.\, \mathbf{A}_o$$

will be written as

$$\forall\, \mathbf{x}, \mathbf{y} : \alpha, \mathbf{z} : \beta \,.\, \mathbf{A}_o.$$

We will also use this form with quasitypes (which are introduced below) (Notational Convention 12).

T_o	stands for	$(\lambda\, x : o \,.\, x) = (\lambda\, x : o \,.\, x)$.
F_o	stands for	$(\lambda\, x : o \,.\, T_o) = (\lambda\, x : o \,.\, x)$.
$\wedge_{o \rightarrow o \rightarrow o}$	stands for	$\lambda\, x : o \,.\, \lambda\, y : o \,.$ $(\lambda\, g : o \rightarrow o \rightarrow o \,.\, g\, T_o\, T_o) =$ $(\lambda\, g : o \rightarrow o \rightarrow o \,.\, g\, x\, y)$.
$(\mathbf{A}_o \wedge \mathbf{B}_o)$	stands for	$\wedge_{o \rightarrow o \rightarrow o}\, \mathbf{A}_o\, \mathbf{B}_o$.
$\Rightarrow_{o \rightarrow o \rightarrow o}$	stands for	$\lambda\, x : o \,.\, \lambda\, y : o \,.\, x = (x \wedge y)$.
$(\mathbf{A}_o \Rightarrow \mathbf{B}_o)$	stands for	$\Rightarrow_{o \rightarrow o \rightarrow o}\; \mathbf{A}_o\, \mathbf{B}_o$.
$\neg_{o \rightarrow o}$	stands for	$\lambda\, x : o \,.\, x = F_o$.
$(\neg \mathbf{A}_o)$	stands for	$\neg_{o \rightarrow o}\, \mathbf{A}_o$.
$\vee_{o \rightarrow o \rightarrow o}$	stands for	$\lambda\, x : o \,.\, \lambda\, y : o \,.\, \neg(\neg x \wedge \neg y)$.
$(\mathbf{A}_o \vee \mathbf{B}_o)$	stands for	$\vee_{o \rightarrow o \rightarrow o}\, \mathbf{A}_o\, \mathbf{B}_o$.

Table 1: Notational Definitions for Boolean Operators

$(\mathbf{A}_\alpha\ \mathbf{c}\ \mathbf{B}_\alpha)$	stands for	$\mathbf{c}_{\alpha \rightarrow \alpha \rightarrow \beta}\, \mathbf{A}_\alpha\, \mathbf{B}_\alpha$ or $\mathbf{c}_{(\alpha \times \alpha) \rightarrow \beta}\, (\mathbf{A}_\alpha, \mathbf{B}_\alpha)$.
$(\mathbf{A}_o \Leftrightarrow \mathbf{B}_o)$	stands for	$\mathbf{A}_o = \mathbf{B}_o$.
$(\mathbf{A}_\alpha \neq \mathbf{B}_\alpha)$	stands for	$\neg(\mathbf{A}_\alpha = \mathbf{B}_\alpha)$.
$(\mathbf{A}_\alpha < \mathbf{B}_\alpha)$	stands for	$(\leq_{\alpha \rightarrow \alpha \rightarrow o}\, \mathbf{A}_\alpha\, \mathbf{B}_\alpha) \wedge (\mathbf{A}_\alpha \neq \mathbf{B}_\alpha)$.
$(\mathbf{A}_\alpha > \mathbf{B}_\alpha)$	stands for	$\mathbf{B}_\alpha < \mathbf{A}_\alpha$.
$(\mathbf{A}_\alpha \geq \mathbf{B}_\alpha)$	stands for	$\mathbf{B}_\alpha \leq \mathbf{A}_\alpha$.
$(\mathbf{A}_\alpha = \mathbf{B}_\alpha = \mathbf{C}_\alpha)$	stands for	$(\mathbf{A}_\alpha = \mathbf{B}_\alpha) \wedge (\mathbf{B}_\alpha = \mathbf{C}_\alpha)$.
$(\mathbf{A}_\alpha\ \mathbf{c}\ \mathbf{B}_\alpha\ \mathbf{d}\ \mathbf{C}_\alpha)$	stands for	$(\mathbf{A}_\alpha\ \mathbf{c}\ \mathbf{B}_\alpha) \wedge (\mathbf{B}_\alpha\ \mathbf{d}\ \mathbf{C}_\alpha)$.

Table 2: Notational Definitions for Binary Operators

In Table 4, we present notation for expressions involving definedness. $\bot_o$ is a canonical "undefined" formula. $\bot_\alpha$ is a canonical undefined expression of type $\alpha \neq o$. $\Delta_{\alpha \rightarrow \beta}$ is the empty function of type $\alpha \rightarrow \beta$ (where $\beta \neq o$). $(\mathbf{A}_\alpha\!\downarrow)$ and $(\mathbf{A}_\alpha\!\uparrow)$ assert that the expression $\mathbf{A}_\alpha$ is defined and undefined, respectively. $(\mathbf{A}_\alpha \simeq \mathbf{B}_\alpha)$ asserts

$(\forall\, \mathbf{x} : \alpha \,.\, \mathbf{A}_o)$	stands for	$(\lambda\, x : \alpha \,.\, T_o) = (\lambda\, \mathbf{x} : \alpha \,.\, \mathbf{A}_o)$.
$(\exists\, \mathbf{x} : \alpha \,.\, \mathbf{A}_o)$	stands for	$\neg(\forall\, \mathbf{x} : \alpha \,.\, \neg \mathbf{A}_o)$.

Table 3: Notational Definitions for Quantifiers

$\bot_o$	stands for	F_o.
$\bot_\alpha$	stands for	$\mathrm{I}\, x : \alpha \,.\, x \neq x$ where $\alpha \neq o$.
$\Delta_{\alpha\to\beta}$	stands for	$\lambda\, x : \alpha \,.\, \bot_\beta$ where $\beta \neq o$.
$(\mathbf{A}_\alpha\downarrow)$	stands for	$\mathbf{A}_\alpha = \mathbf{A}_\alpha$.
$(\mathbf{A}_\alpha\uparrow)$	stands for	$\neg(\mathbf{A}_\alpha\downarrow)$.
$(\mathbf{A}_\alpha \simeq \mathbf{B}_\alpha)$	stands for	$(\mathbf{A}_\alpha\downarrow \vee \mathbf{B}_\alpha\downarrow) \Rightarrow \mathbf{A}_\alpha = \mathbf{B}_\alpha$.
$(\mathbf{A}_\alpha \not\simeq \mathbf{B}_\alpha)$	stands for	$\neg(\mathbf{A}_\alpha \simeq \mathbf{B}_\alpha)$.
$\mathsf{IF}(\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o)$	stands for	$(\mathbf{A}_o \Rightarrow \mathbf{B}_o) \wedge (\neg\mathbf{A}_o \Rightarrow \mathbf{C}_o)$.
$\mathsf{IF}(\mathbf{A}_o, \mathbf{B}_\alpha, \mathbf{C}_\alpha)$	stands for	$\mathrm{I}\, x : \alpha \,.$ $(\mathbf{A}_o \Rightarrow x = \mathbf{B}_\alpha) \wedge (\neg\mathbf{A}_o \Rightarrow x = \mathbf{C}_\alpha)$ where $\alpha \neq o$.
$(\mathbf{A}_o \mapsto \mathbf{B}_\alpha \mid \mathbf{C}_\alpha)$	stands for	$\mathsf{IF}(\mathbf{A}_o, \mathbf{B}_\alpha, \mathbf{C}_\alpha)$.

Table 4: Notational Definitions for Definedness

that the expressions $\mathbf{A}_\alpha$ and $\mathbf{B}_\alpha$ are *quasi-equal*, i.e., they are both defined and equal or both undefined. And $(\mathbf{A}_o \mapsto \mathbf{B}_\alpha \mid \mathbf{C}_\alpha)$ is a conditional expression that denotes the value of $\mathbf{B}_\alpha$ if $\mathbf{A}_o$ holds and otherwise denotes the value of $\mathbf{C}_\alpha$.

The notation $\bot_\alpha$ is an example of a *parametric pseudoconstant*. It stands for an expression $\mathbf{C}_\alpha$ where α is a *parametric type* with the syntactic variable α serving as a parameter that can be freely replaced with any type. Thus $\bot_\alpha$ is polymorphic in the sense that it can be used with expressions of different types by simply replacing the syntactic variable α with the type that is needed. $\Delta_{\alpha\to\beta}$ is similarly a parametric pseudoconstant.

The notational definitions of $\mathsf{IF}(\mathbf{A}_o, \mathbf{B}_o, \mathbf{C}_o)$ and $\mathsf{IF}(\mathbf{A}_o, \mathbf{B}_\alpha, \mathbf{C}_\alpha)$ (where $\alpha \neq o$) are *(parameterized) abbreviations* of the form

$$A(\mathbf{B}^1_{\alpha_1}, \ldots, \mathbf{B}^n_{\alpha_n}) \text{ stands for } C$$

where A is a name, $n \geq 0$, and the syntactic variables $\mathbf{B}^1_{\alpha_1}, \ldots, \mathbf{B}^1_{\alpha_1}$ appear in the expression C. A is written in uppercase sans sarif font to distinguish it from the name of a constant or pseudoconstant (Notational Convention 13). We will always assume that the bound variables introduced in C are chosen so that they are not free in $\mathbf{B}^1_{\alpha_1}, \ldots, \mathbf{B}^1_{\alpha_1}$(Notational Convention 14). For example, the bound variable $(x : \alpha)$ in the RHS of the notational definition of $\mathsf{IF}(\mathbf{A}_o, \mathbf{B}_\alpha, \mathbf{C}_\alpha)$ (where $\alpha \neq o$) in Table 4 is chosen so that it is not free in $\mathbf{A}_o$, $\mathbf{B}_\alpha$, or $\mathbf{C}_\alpha$.

Since we can identify a set $S \subseteq U$ with the predicate $p_S : U \to \mathbb{B}$ such that $a \in S$ iff $p_S(a)$, we will introduce a power set type of α, i.e., a type of the subsets of

$\{\alpha\}$	stands for	$\alpha \rightarrow o$.
$(\mathbf{A}_\alpha \in \mathbf{B}_{\{\alpha\}})$	stands for	$\mathbf{B}_{\{\alpha\}}\, \mathbf{A}_\alpha$.
$(\mathbf{A}_\alpha \not\in \mathbf{B}_{\{\alpha\}})$	stands for	$\neg(\mathbf{A}_\alpha \in \mathbf{B}_{\{\alpha\}})$.
$\{\mathbf{x} : \alpha \mid \mathbf{A}_o\}$	stands for	$\lambda\, \mathbf{x} : \alpha \,.\, \mathbf{A}_o$.
$\emptyset_{\{\alpha\}}$	stands for	$\lambda\, x : \alpha \,.\, F_o$.
$\{\,\}_{\{\alpha\}}$	stands for	$\emptyset_{\{\alpha\}}$.
$U_{\{\alpha\}}$	stands for	$\lambda\, x : \alpha \,.\, T_o$.
n-α-SET	stands for	$\lambda\, x_1 : \alpha \,.\, \cdots \,.\, \lambda\, x_n : \alpha \,.\, \lambda\, x : \alpha \,.$ $x = x_1 \vee \cdots \vee x = x_n$ where $n \geq 1$.
$\{\mathbf{A}^1_\alpha, \ldots, \mathbf{A}^n_\alpha\}$	stands for	n-α-$\mathsf{SET}\, \mathbf{A}^1_\alpha \cdots \mathbf{A}^n_\alpha$ where $n \geq 1$.
$\subseteq_{\{\alpha\} \rightarrow \{\alpha\} \rightarrow o}$	stands for	$\lambda\, a : \{\alpha\} \,.\, \lambda\, b : \{\alpha\} \,.$ $\forall\, x : \alpha \,.\, x \in a \Rightarrow x \in b$.
$\cup_{\{\alpha\} \rightarrow \{\alpha\} \rightarrow \{\alpha\}}$	stands for	$\lambda\, a : \{\alpha\} \,.\, \lambda\, b : \{\alpha\} \,.$ $\{x : \alpha \mid x \in a \vee x \in b\}$.
$\cap_{\{\alpha\} \rightarrow \{\alpha\} \rightarrow \{\alpha\}}$	stands for	$\lambda\, a : \{\alpha\} \,.\, \lambda\, b : \{\alpha\} \,.$ $\{x : \alpha \mid x \in a \wedge x \in b\}$.
$\overline{\cdot}_{\{\alpha\} \rightarrow \{\alpha\}}$	stands for	$\lambda\, a : \{\alpha\} \,.\, \{x : \alpha \mid x \not\in a\}$.
$\overline{\mathbf{A}_{\{\alpha\}}}$	stands for	$\overline{\cdot}_{\{\alpha\} \rightarrow \{\alpha\}}\, \mathbf{A}_{\{\alpha\}}$.
$\setminus_{\{\alpha\} \rightarrow \{\alpha\} \rightarrow \{\alpha\}}$	stands for	$\lambda\, a : \{\alpha\} \,.\, \lambda\, b : \{\alpha\} \,.\, a \cap \overline{b}$.

Table 5: Notational Definitions for Sets

(α)	stands for	α.
$(\alpha_1 \times \cdots \times \alpha_n)$	stands for	$(\alpha_1 \times (\alpha_2 \times \cdots \times \alpha_n))$ where $n \geq 2$.
$(\mathbf{A}_\alpha)$	stands for	$\mathbf{A}_\alpha$.
$(\mathbf{A}^1_{\alpha_1}, \ldots, \mathbf{A}^n_{\alpha_n})$	stands for	$(\mathbf{A}^1_{\alpha_1}, (\mathbf{A}^2_{\alpha_1}, \ldots, \mathbf{A}^n_{\alpha_n}))$ where $n \geq 2$.
$\mathsf{fst}_{(\alpha \times \beta) \rightarrow \alpha}$	stands for	$\lambda\, p : \alpha \times \beta \,.\, \mathrm{I}\, x : \alpha \,.\, \exists\, y : \beta \,.\, p = (x, y)$.
$\mathsf{snd}_{(\alpha \times \beta) \rightarrow \beta}$	stands for	$\lambda\, p : \alpha \times \beta \,.\, \mathrm{I}\, y : \beta \,.\, \exists\, x : \alpha \,.\, p = (x, y)$.

Table 6: Notational Definitions for Tuples

α, as the type $\alpha \rightarrow o$ of predicates on α. The compact notation for $\alpha \rightarrow o$ is $\{\alpha\}$. We introduce this notation and compact notation for the common set operators in Table 5. $\emptyset_{\{\alpha\}}$ and $U_{\{\alpha\}}$ are parametric pseudoconstants that denote the empty set and the universal set, respectively, of the members in the domain of α.

We introduce notation for product types, tuples, and the accessors for ordered pairs in Table 6.

$\mathsf{id}_{\alpha\to\alpha}$	stands for	$\lambda\, x : \alpha \,.\, x.$
$\mathsf{dom}_{(\alpha\to\beta)\to\{\alpha\}}$	stands for	$\lambda\, f : \alpha \to \beta \,.$ $\{x : \alpha \mid (f\, x)\downarrow\}.$
$\mathsf{ran}_{(\alpha\to\beta)\to\{\beta\}}$	stands for	$\lambda\, f : \alpha \to \beta \,.$ $\{y : \beta \mid \exists\, x : \alpha \,.\, f\, x = y\}.$
$\mathsf{TOTAL}(\mathbf{F}_{\alpha\to\beta})$	stands for	$\forall\, x : \alpha \,.\, (\mathbf{F}_{\alpha\to\beta}\, x)\downarrow.$
$\vert_{(\alpha\to\beta)\to\{\alpha\}\to(\alpha\to\beta)}$	stands for	$\lambda\, f : \alpha \to \beta \,.\, \lambda\, s : \{\alpha\} \,.$ $\lambda\, x : \alpha \,.\, x \in s \mapsto f\, x \mid \bot_\beta.$
$(\mathbf{F}_{\alpha\to\beta}\vert_{\mathbf{A}_{\{\alpha\}}})$	stands for	$\vert_{(\alpha\to\beta)\to\{\alpha\}\to(\alpha\to\beta)}\, \mathbf{F}_{\alpha\to\beta}\, \mathbf{A}_{\{\alpha\}}.$

Table 7: Notational Definitions for Functions

Some convenient notation for functions is found in Table 7.

A *quasitype within type* $\alpha \in \mathcal{T}$ is any expression of type $\{\alpha\} = \alpha \to o$. A quasitype $\mathbf{Q}_{\{\alpha\}}$ denotes a subset of the domain denoted by α. Thus quasitypes represent subtypes and are useful for specifying subdomains of a domain. Unlike a type, a quasitype may denote an empty domain. Notice that an expression $\mathbf{A}_{\alpha\to o}$ is simultaneously an expression of type $\alpha \to o$, an expression of type of $\{\alpha\}$, and a quasitype within type α. So $\mathbf{A}_{\alpha\to o}$ (or $\mathbf{A}_{\{\alpha\}}$) can be used as a function, as a set, and like a type as shown below.

In Table 8, we introduce various notations for using quasitypes in place of types. Quasitypes can be used to restrict the range of a variable bound by a binder. For example, $(\lambda\, x : \mathbf{Q}_{\{\alpha\}} \,.\, \mathbf{B}_\beta)$ denotes the function denoted by $\lambda\, x : \alpha \,.\, \mathbf{B}_\beta$ weakly restricted to the domain denoted by $\mathbf{Q}_{\{\alpha\}}$. Quasitypes can also be used to sharpen definedness statements. For example, $(\mathbf{A}_\alpha \downarrow \mathbf{Q}_{\{\alpha\}})$, read as $\mathbf{A}_\alpha$ *is defined in* $\mathbf{Q}_{\{\alpha\}}$, asserts that the value of $\mathbf{A}_\alpha$ is defined and is a member of the set denoted by $\mathbf{Q}_{\{\alpha\}}$. $(\mathbf{Q}_{\{\alpha\}} \to \mathbf{R}_{\{\beta\}})$ is a quasitype within $\alpha \to \beta$ that denotes the function space from the denotation of $\mathbf{Q}_{\{\alpha\}}$ to the denotation of $\mathbf{R}_{\{\beta\}}$, and $(\mathbf{Q}_{\{\alpha\}} \times \mathbf{R}_{\{\beta\}})$ is a quasitype within $\alpha \times \beta$ that denotes the Cartesian product of the denotation of $\mathbf{Q}_{\{\alpha\}}$ and the denotation of $\mathbf{R}_{\{\beta\}}$.

4 Monoids

A *monoid* is a mathematical structure $(m, \cdot, e)$ where m is a nonempty set of values, $\cdot : (m \times m) \to m$ is an associative function, and $e \in m$ is an identity element with respect to $\cdot$. Mathematics and computing are replete with examples of monoids such as $(\mathbb{N}, +, 0)$, $(\mathbb{N}, *, 1)$, and $(\Sigma^*, {+}{+}, \epsilon)$ where Σ^* is the set of strings over an alphabet

$(\lambda\, \mathbf{x} : \mathbf{Q}_{\{\alpha\}} \,.\, \mathbf{B}_\beta)$	stands for	$\lambda\, \mathbf{x} : \alpha \,.\, (\mathbf{x} \in \mathbf{Q}_{\{\alpha\}} \mapsto \mathbf{B}_\beta \mid \bot_\beta)$.
$(\forall\, \mathbf{x} : \mathbf{Q}_{\{\alpha\}} \,.\, \mathbf{B}_o)$	stands for	$\forall\, \mathbf{x} : \alpha \,.\, (\mathbf{x} \in \mathbf{Q}_{\{\alpha\}} \Rightarrow \mathbf{B}_o)$.
$(\exists\, \mathbf{x} : \mathbf{Q}_{\{\alpha\}} \,.\, \mathbf{B}_o)$	stands for	$\exists\, \mathbf{x} : \alpha \,.\, (\mathbf{x} \in \mathbf{Q}_{\{\alpha\}} \wedge \mathbf{B}_o)$.
$(\mathrm{I}\, \mathbf{x} : \mathbf{Q}_{\{\alpha\}} \,.\, \mathbf{B}_o)$	stands for	$\mathrm{I}\, \mathbf{x} : \alpha \,.\, (\mathbf{x} \in \mathbf{Q}_{\{\alpha\}} \wedge \mathbf{B}_o)$.
$(\mathbf{A}_\alpha \downarrow \mathbf{Q}_{\{\alpha\}})$	stands for	$\mathbf{A}_\alpha\!\downarrow \wedge\, \mathbf{A}_\alpha \in \mathbf{Q}_{\{\alpha\}}$.
$(\mathbf{A}_\alpha \uparrow \mathbf{Q}_{\{\alpha\}})$	stands for	$\neg(\mathbf{A}_\alpha \downarrow \mathbf{Q}_{\{\alpha\}})$.
$\rightarrow_{\{\alpha\}\to\{\beta\}\to\{\alpha\to\beta\}}$	stands for	$\lambda\, s : \{\alpha\} \,.\, \lambda\, t : \{\beta\} \,.$
		$\{f : \alpha \to \beta \mid \forall\, x : \alpha \,.$
		$(f\, x)\!\downarrow \Rightarrow (x \in s \wedge f\, x \in t)\}$
		where $\beta \neq o$.
$\times_{\{\alpha\}\to\{\beta\}\to\{\alpha\times\beta\}}$	stands for	$\lambda\, s : \{\alpha\} \,.\, \lambda\, t : \{\beta\} \,.$
		$\{p : \alpha \times \beta \mid$
		$\mathsf{fst}_{(\alpha\times\beta)\to\alpha}\, p \in s\, \wedge$
		$\mathsf{snd}_{(\alpha\times\beta)\to\beta}\, p \in t\}$
$(\mathbf{Q}_{\{\alpha\}} \to o)$	stands for	$\{s : \{\alpha\} \mid s \subseteq \mathbf{Q}_{\{\alpha\}}\}$.
$\mathcal{P}(\mathbf{Q}_{\{\alpha\}})$	stands for	$\mathbf{Q}_{\{\alpha\}} \to o$.
$(\mathbf{Q}_{\{\alpha\}} \to \mathbf{R}_{\{\beta\}})$	stands for	$\rightarrow_{\{\alpha\}\to\{\beta\}\to\{\alpha\to\beta\}}\ \mathbf{Q}_{\{\alpha\}}\, \mathbf{R}_{\{\beta\}}$
		where $\beta \neq o$.
$(\alpha \to \mathbf{R}_{\{\beta\}})$	stands for	$U_{\{\alpha\}} \to \mathbf{R}_{\{\beta\}}$ where $\beta \neq o$.
$(\mathbf{Q}_{\{\alpha\}} \to \beta)$	stands for	$\mathbf{Q}_{\{\alpha\}} \to U_{\{\beta\}}$ where $\beta \neq o$.
$(\mathbf{Q}_{\{\alpha\}} \times \mathbf{R}_{\{\beta\}})$	stands for	$\times_{\{\alpha\}\to\{\beta\}\to\{\alpha\times\beta\}}\, \mathbf{Q}_{\{\alpha\}}\, \mathbf{R}_{\{\beta\}}$.
$(\alpha \times \mathbf{R}_{\{\beta\}})$	stands for	$U_{\{\alpha\}} \times \mathbf{R}_{\{\beta\}}$.
$(\mathbf{Q}_{\{\alpha\}} \times \beta)$	stands for	$\mathbf{Q}_{\{\alpha\}} \times U_{\{\beta\}}$.
$\mathsf{TOTAL\text{-}ON}(\mathbf{F}_{\alpha\to\beta}, \mathbf{Q}_{\{\alpha\}}, \mathbf{R}_{\{\beta\}})$	stands for	$\forall\, x : \mathbf{Q}_{\{\alpha\}} \,.\, (\mathbf{F}_{\alpha\to\beta}\, x) \downarrow \mathbf{R}_{\{\beta\}}$.

Table 8: Notational Definitions for Quasitypes

Σ, $++$ is string concatenation, and ϵ is the empty string.

Table 9 defines some parametric pseudoconstants that we will need for monoids, and Table 10 defines several useful abbreviations for monoids.

Let $T = (L, \Gamma)$ be a theory[3] of Alonzo. Consider a tuple

$$(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$$

where (1) ζ_α is either a type α of L or a closed quasitype $\mathbf{Q}_{\{\alpha\}}$ of L and (2) $\mathbf{F}_{(\alpha\times\alpha)\to\alpha}$ and $\mathbf{E}_\alpha$ are closed expressions of L. Let $\mathbf{X}_o$ be the sentence

$$\mathsf{MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha),$$

where MONOID is the abbreviation introduced by the notational definition given in Table 10 and $\mathbf{M}_{\{\alpha\}}$ is $U_{\{\alpha\}}$ if ζ_α is α and is $\mathbf{Q}_{\{\alpha\}}$ otherwise. If $T \vDash \mathbf{X}_o$, then $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ denotes a monoid $(m, \cdot, e)$ in T. Stated more precisely,

[3] A *theory* of Alonzo and related notions are presented in Chapter 9 of [20].

$\mathsf{set\text{-}op}_{((\alpha\times\beta)\to\gamma)\to((\{\alpha\}\times\{\beta\})\to\{\gamma\})}$ stands for $\lambda\, f : (\alpha \times \beta) \to \gamma \,.\, \lambda\, p : \{\alpha\} \times \{\beta\}\,.$ $\{z : \gamma \mid \exists\, x : \mathsf{fst}\, p,\, y : \mathsf{snd}\, p \,.\, z = f\,(x, y)\}.$
$\circ_{((\alpha\to\beta)\times(\beta\to\gamma))\to(\alpha\to\gamma)}$ stands for $\lambda\, p : (\alpha \to \beta) \times (\beta \to \gamma) \,.\, \lambda\, x : \alpha \,.\, (\mathsf{snd}\, p)\, ((\mathsf{fst}\, p)\, x).$
$(\mathbf{F}_{\alpha\to\beta} \circ \mathbf{G}_{\beta\to\gamma})$ stands for $\circ_{((\alpha\to\beta)\times(\beta\to\gamma))\to(\alpha\to\gamma)}\, (\mathbf{F}_{\alpha\to\beta}, \mathbf{G}_{\beta\to\gamma}).$
$\bullet_{((\alpha\to\beta)\times\alpha)\to\beta}$ stands for $\lambda\, p : (\alpha \to \beta) \times \alpha \,.\, (\mathsf{fst}\, p)\, (\mathsf{snd}\, p).$

Table 9: Notational Definitions for Monoids: Pseudoconstants

if $T \vDash \mathbf{X}_o$, then, for all general models M of T and all assignments $\varphi \in \mathsf{assign}(M)$, $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ denotes the monoid

$$(V^M_\varphi(\mathbf{M}_{\{\alpha\}}), V^M_\varphi(\mathbf{F}_{(\alpha\times\alpha)\to\alpha}), V^M_\varphi(\mathbf{E}_\alpha)).$$

Thus we can show that $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ denotes a monoid in T by proving $T \vDash \mathbf{X}_o$. However, we may need general definitions and theorems about monoids to prove properties in T about the monoid denoted by $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$. It would be extremely inefficient to state these definitions and prove these theorems in T since instances of these same definitions and theorems could easily be needed for other triples in T, as well as in other theories, that denote monoids.

Instead of developing part of a monoid theory in T, we should apply the little theories method and develop a "little theory" T_{mon} of monoids, separate from T, that has the most convenient level of abstraction and the most convenient vocabulary for talking about monoids. The general definitions and theorems of monoids can then be introduced in a development[4] D_{mon} of T_{mon} in a universal abstract form. When these

[4] A *development* of Alonzo and related notions are presented in Chapter 12 of [20].

$\mathsf{MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ stands for $\mathbf{M}_{\{\alpha\}}\downarrow \wedge$ $\mathbf{M}_{\{\alpha\}} \neq \emptyset_{\{\alpha\}} \wedge$ $\mathbf{F}_{(\alpha\times\alpha)\to\alpha} \downarrow (\mathbf{M}_{\{\alpha\}} \times \mathbf{M}_{\{\alpha\}}) \to \mathbf{M}_{\{\alpha\}} \wedge$ $\mathbf{E}_\alpha \downarrow \mathbf{M}_{\{\alpha\}} \wedge$ $\forall\, x, y, z : \mathbf{M}_{\{\alpha\}}\ .$ $\quad \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(x, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(y, z)) = \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(\mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(x, y), z) \wedge$ $\forall\, x : \mathbf{M}_{\{\alpha\}}\ .\ \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(\mathbf{E}_\alpha, x) = \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(x, \mathbf{E}_\alpha) = x.$
$\mathsf{COM\text{-}MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ stands for $\mathsf{MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha) \wedge$ $\forall\, x, y : \mathbf{M}_{\{\alpha\}}\ .\ \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(x, y) = \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(y, x)$
$\mathsf{MON\text{-}ACTION}(\mathbf{M}_{\{\alpha\}}, \mathbf{S}_{\{\beta\}}, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha, \mathbf{G}_{(\alpha\times\beta)\to\beta})$ stands for $\mathsf{MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha) \wedge$ $\mathbf{S}_{\{\beta\}}\downarrow \wedge$ $\mathbf{S}_{\{\beta\}} \neq \emptyset_{\{\beta\}} \wedge$ $\mathbf{G}_{(\alpha\times\beta)\to\beta} \downarrow (\mathbf{M}_{\{\alpha\}} \times \mathbf{S}_{\{\beta\}}) \to \mathbf{S}_{\{\beta\}} \wedge$ $\forall\, x, y : \mathbf{M}_{\{\alpha\}},\ s : \mathbf{S}_{\{\beta\}}\ .$ $\quad \mathbf{G}_{(\alpha\times\beta)\to\beta}\,(x, \mathbf{G}_{(\alpha\times\beta)\to\beta}\,(y, s)) = \mathbf{G}_{(\alpha\times\beta)\to\beta}\,(\mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(x, y), s) \wedge$ $\forall\, s : \mathbf{S}_{\{\beta\}}\ .\ \mathbf{G}_{(\alpha\times\beta)\to\beta}\,(\mathbf{E}_\alpha, s) = s.$
$\mathsf{MON\text{-}HOMOM}(\mathbf{M}^1_{\{\alpha\}}, \mathbf{M}^2_{\{\beta\}}, \mathbf{F}^1_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}^1_\alpha, \mathbf{F}^2_{(\beta\times\beta)\to\beta}, \mathbf{E}^2_\beta, \mathbf{H}_{\alpha\to\beta})$ stands for $\mathsf{MONOID}(\mathbf{M}^1_{\{\alpha\}}, \mathbf{F}^1_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}^1_\alpha) \wedge$ $\mathsf{MONOID}(\mathbf{M}^2_{\{\beta\}}, \mathbf{F}^2_{(\beta\times\beta)\to\beta}, \mathbf{E}^2_\beta) \wedge$ $\mathbf{H}_{\alpha\to\beta} \downarrow \mathbf{M}^1_{\{\alpha\}} \to \mathbf{M}^2_{\{\beta\}} \wedge$ $\forall\, x, y : \mathbf{M}^1_{\{\alpha\}}\ .\ \mathbf{H}_{\alpha\to\beta}\,(\mathbf{F}^1_{(\alpha\times\alpha)\to\alpha}\,(x, y)) = \mathbf{F}^2_{(\beta\times\beta)\to\beta}\,(\mathbf{H}_{\alpha\to\beta}\, x, \mathbf{H}_{\alpha\to\beta}\, y) \wedge$ $\mathbf{H}_{\alpha\to\beta}\, \mathbf{E}^1_\alpha = \mathbf{E}^2_\beta$

Table 10: Notational Definitions for Monoids: Abbreviations

definitions and theorems are needed in a development D, a development morphism[5] from D_{mon} to D can be created and then used to transport the abstract definitions and theorems in D_{mon} to concrete instances of them in D. The validity of these concrete definitions and theorems in D is guaranteed by the fact that the abstract definitions and theorems are valid in the top theory of D_{mon} and the development morphism used to transport them preserves validity.

We can verify that $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ denotes a monoid in T by simply constructing an appropriate theory morphism Φ from T_{mon} to T. As a bonus, we can use Φ to transport the abstract definitions and theorems in D_{mon} to concrete instances of them in a development of T whenever they are needed. Moreover, we do not have to explicitly prove that a particular property of $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$, such as $\mathbf{X}_o$, that holds by virtue of $(\zeta_\alpha, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}, \mathbf{E}_\alpha)$ denoting a monoid is valid in T; instead, we only need to show that there is an abstract theorem of T_{mon} that Φ transports to this property.

The following theory definition module defines a suitably abstract theory of monoids named MON:

Theory Definition 4.1 (Monoids)**.**

Name: MON.

Base types: M.

Constants: $\cdot_{(M\times M)\to M}$, e_M.

Axioms:

1. $\forall x, y, z : M \,.\, x \cdot (y \cdot z) = (x \cdot y) \cdot z$ ($\cdot$ is associative).

2. $\forall x : M \,.\, \mathsf{e} \cdot x = x \cdot \mathsf{e} = x$ (e is an identity element with respect to $\cdot$).

Notice that we have employed several notational definitions and conventions in the axioms — including dropping the types of the constants — for the sake of brevity. This theory specifies the set of monoids exactly: The base type M, like all types, denotes a nonempty set m; the constant $\cdot_{(M\times M)\to M}$ denotes a function $\cdot : (m \times m) \to m$ that is associative; and the constant e_M denotes a member e of m that is an identity element with respect to $\cdot$.

The following development definition module defines a development, named MON-1, of the theory MON:

[5] A *theory morphism* and a *development morphism* of Alonzo are presented in Sections 14.3 and 14.4, respectively, of [20].

Development Definition 4.2 (Monoids 1)**.**

Name: MON-1.

Bottom theory: MON.

Definitions and theorems:

Thm1: $\mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \rightarrow M}, \mathsf{e}_M)$
(models of MON define monoids).

Thm2: $\mathsf{TOTAL}(\cdot_{(M \times M) \rightarrow M})$ ($\cdot$ is total).

Thm3: $\forall\, x : M \,.\, (\forall\, y : M \,.\, x \cdot y = y \cdot x = y) \Rightarrow x = \mathsf{e}$
(uniqueness of identity element).

Def1: $\mathsf{submonoid}_{\{M\} \rightarrow o} =$
$\lambda\, s : \{M\} \,.\, s \neq \emptyset_{\{M\}} \wedge (\cdot|_{s \times s} \downarrow (s \times s) \rightarrow s) \wedge \mathsf{e} \in s$ (submonoid).

Thm4: $\forall\, s : \{M\} \,.\, \mathsf{submonoid}\, s \Rightarrow \mathsf{MONOID}(s, \cdot|_{s \times s}, \mathsf{e})$
(submonoids are monoids).

Thm5: $\mathsf{submonoid}\, \{\mathsf{e}\}$ (minimum submonoid).

Thm6: $\mathsf{submonoid}\, U_{\{M\}}$ (maximum submonoid).

Def2: $\cdot^{\mathrm{op}}_{(M \times M) \rightarrow M} = \lambda\, p : M \times M \,.\, (\mathsf{snd}\, p) \cdot (\mathsf{fst}\, p)$ (opposite of $\cdot$).

Thm7: $\forall\, x, y, z : M \,.\, x \cdot^{\mathrm{op}} (y \cdot^{\mathrm{op}} z) = (x \cdot^{\mathrm{op}} y) \cdot^{\mathrm{op}} z$
($\cdot^{\mathrm{op}}$ is associative).

Thm8: $\forall\, x : M \,.\, \mathsf{e} \cdot^{\mathrm{op}} x = x \cdot^{\mathrm{op}} \mathsf{e} = x$
(e is an identity element with respect to $\cdot^{\mathrm{op}}$).

Def3: $\odot_{(\{M\} \times \{M\}) \rightarrow \{M\}} = \mathsf{set\text{-}op}_{((M \times M) \rightarrow M) \rightarrow ((\{M\} \times \{M\}) \rightarrow \{M\})}\, \cdot$
(set product).

Def4: $\mathsf{E}_{\{M\}} = \{\mathsf{e}_M\}$ (set identity element).

Thm9: $\forall\, x, y, z : \{M\} \,.\, x \odot (y \odot z) = (x \odot y) \odot z$ ($\odot$ is associative).

Thm10: $\forall\, x : \{M\} \,.\, \mathsf{E} \odot x = x \odot \mathsf{E} = x$
(E is an identity element with respect to $\odot$).

$\mathsf{set\text{-}op}_{((M \times M) \rightarrow M) \rightarrow ((\{M\} \times \{M\}) \rightarrow \{M\})}$ is an instance of the parametric pseudoconstant $\mathsf{set\text{-}op}_{((\alpha \times \beta) \rightarrow \gamma) \rightarrow ((\{\alpha\} \times \{\beta\}) \rightarrow \{\gamma\})}$ defined in Table 9.

Thm1 states that each model of MON defines a monoid. Thm2 states that the monoid's binary function is total (which is implied by the first axiom of MON). Thm3 states that a monoid's identity element is unique. Def1 defines the notion of a *submonoid* and Thm4–Thm6 are three theorems about submonoids. Notice that $\cdot|_{s \times s}$, the restriction of $\cdot$ to $s \times s$, denotes a partial function. Notice also that

$$\cdot|_{s \times s} \downarrow (s \times s) \rightarrow s$$

in Def1 asserts that s is closed under $\cdot|_{s \times s}$ since $\cdot$ is total by Thm2. Def2 defines $\cdot^{\mathrm{op}}_{(M \times M) \rightarrow M}$, the *opposite* of $\cdot$, and Thm7–Thm8 are key theorems about $\cdot^{\mathrm{op}}$. Def3 defines $\odot_{(\{M\} \times \{M\}) \rightarrow \{M\}}$, the *set product* on $\{M\}$; Def4 defines $\mathsf{E}_{\{M\}}$, the identity element with respect to $\odot$; and Thm9–Thm10 are key theorems about $\odot$. These four definitions and ten theorems require proofs that show the RHS of each definition (i.e., the definition's definiens) is defined and each theorem is valid. The proofs are given in Appendix A.

5 Transportation of definitions and theorems

Let T be a theory such that $T \vDash \mathbf{X}_o$ where $\mathbf{X}_o$ is the sentence

$$\mathsf{MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}, \mathbf{E}_\alpha),$$

and assume that D is some development of T (which could be T itself). We would like to show how the definitions and theorems of the development MON-1 can be transported to D.[6]

Before considering the general case, we will consider the special case when $\mathbf{M}_{\{\alpha\}}$ is $U_{\{\alpha\}}$, which denotes the entire domain for the type α, and $\mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}$ and $\mathbf{E}_\alpha$ are constants $\mathbf{c}_{(\alpha \times \alpha) \rightarrow \alpha}$ and $\mathbf{d}_\alpha$. We start by defining a theory morphism from MON to T using a theory translation definition module:

Theory Translation Definition 5.1 (Special MON to T)**.**

Name: special-MON-to-T.

Source theory: MON.

Target theory: T.

Base type mapping:

[6] A *transportation* is presented in Subsection 14.4.2 of [20].

1. $M \mapsto \alpha$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \mathbf{c}_{(\alpha \times \alpha) \to \alpha}$.
2. $\mathsf{e}_M \mapsto \mathbf{d}_\alpha$.

Since special-MON-to-T is a normal translation[7], it has no obligations of the first kind by [20, Lemma 14.10] and two obligations of the second kind which are valid in T by [20, Lemma 14.11]. It has two obligations of the third kind corresponding to the two axioms of MON. $T \vDash \mathbf{X}_o$ implies that each of these two obligations is valid in T. Therefore, special-MON-to-T is a theory morphism from MON to T by the Morphism Theorem [20, Theorem 14.16].[8]

Now we can transport the definitions and theorems of MON-1 to D via special-MON-to-T using definition and theorems transportation modules. For example, Thm3 and Def1 can be transported using the following two modules:

Theorem Transportation 5.2 (Transport of Thm3 to D)**.**

Name: uniqueness-of-identity-element-via-special-MON-to-D.

Source development: MON-1.

Target development: D.

Development morphism: special-MON-to-T.

Theorem:

Thm3: $\forall\, x : M \,.\, (\forall\, y : M \,.\, x \cdot y = y \cdot x = y) \Rightarrow x = \mathsf{e}$
(uniqueness of identity element).

Transported theorem:

Thm3-via-special-MON-to-T:
$\forall\, x : \alpha \,.\, (\forall\, y : \alpha \,.\, x \;\mathbf{c}\; y = y \;\mathbf{c}\; x = y) \Rightarrow x = \mathbf{d}$
(uniqueness of identity element).

New target development: D'.

[7] A *theory translation* and a *development translation* of Alonzo are presented in Subsections 14.3.1 and 14.4.1, respectively, of [20].

[8] An *obligation* of a theory translation and the Morphism Theorem are presented in Subsection 14.3.2 of [20].

Definition Transportation 5.3 (Transport of Def1 to D')**.**

Name: submonoid-via-special-MON-to-D'.

Source development: MON-1.

Target development: D'.

Development morphism: special-MON-to-T.

Definition:

Def1: $\mathsf{submonoid}_{\{M\} \rightarrow o} =$
$\lambda\, s : \{M\} \,.\, s \neq \emptyset_{\{M\}} \wedge (\cdot|_{s \times s} \downarrow (s \times s) \rightarrow s) \wedge \mathsf{e} \in s$ (submonoid).

Transported definition:

Def1-via-special-MON-to-T: $\mathsf{submonoid}_{\{\alpha\} \rightarrow o} =$
$\lambda\, s : \{\alpha\} \,.\, s \neq \emptyset_{\{\alpha\}} \wedge (\mathbf{c}|_{s \times s} \downarrow (s \times s) \rightarrow s) \wedge \mathbf{d} \in s$ (submonoid).

New target development: D''.

New development morphism: special-MON-1-to-D'.

We will next consider the general case when $\mathbf{M}_{\{\alpha\}}$ may be different from $U_{\{\alpha\}}$ and $\mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}$ and $\mathbf{E}_\alpha$ may not be constants. The general case is usually more complicated and less succinct than the special case. We start again by defining a theory morphism from MON to T using a theory translation definition module:

Theory Translation Definition 5.4 (General MON to T)**.**

Name: general-MON-to-T.

Source theory: MON.

Target theory: T.

Base type mapping:

1. $M \mapsto \mathbf{M}_{\{\alpha\}}$.

Constant mapping:

1. $\cdot_{(M \times M) \rightarrow M} \mapsto \mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}$.
2. $\mathsf{e}_M \mapsto \mathbf{E}_\alpha$.

Let general-MON-to-$T = (\mu, \nu)$. Then general-MON-to-T has the following five obligations (one of the first, two of the second, and two of the third kind):

1. $\overline{\nu}(U_{\{M\}} \neq \emptyset_{\{M\}}) \equiv (\lambda\, x : \mathbf{M}_{\{\alpha\}} \,.\, T_o) \neq (\lambda\, x : \mathbf{M}_{\{\alpha\}} \,.\, F_o)$.
2. $\overline{\nu}(\cdot_{(M \times M) \rightarrow M} \downarrow U_{\{(M \times M) \rightarrow M\}}) \equiv$
   $\mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha} \downarrow (\lambda\, x : (\mathbf{M}_{\{\alpha\}} \times \mathbf{M}_{\{\alpha\}}) \rightarrow \mathbf{M}_{\{\alpha\}} \,.\, T_o)$.
3. $\overline{\nu}(\mathsf{e}_M \downarrow U_{\{M\}}) \equiv \mathbf{E}_\alpha \downarrow (\lambda\, x : \mathbf{M}_{\{\alpha\}} \,.\, T_o)$.
4. $\overline{\nu}(\forall\, x, y, z : M \,.\, x \cdot (y \cdot z) = (x \cdot y) \cdot z) \equiv$
   $\forall\, x, y, z : \mathbf{M}_{\{\alpha\}} \,.$
   $\mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}(x, \mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}(y, z)) = \mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}(\mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}(x, y), z)$.
5. $\overline{\nu}(\forall\, x : M \,.\, \mathsf{e} \cdot x = x \cdot \mathsf{e} = x) \equiv$
   $\forall\, x : \mathbf{M}_{\{\alpha\}} \,.\, \mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}(\mathbf{E}_\alpha, x) = \mathbf{F}_{(\alpha \times \alpha) \rightarrow \alpha}(x, \mathbf{E}_\alpha) = x$.

$\mathbf{A}_\alpha \equiv \mathbf{B}_\alpha$ means the expressions denoted by $\mathbf{A}_\alpha$ and $\mathbf{B}_\alpha$ are identical.

$T \vDash \mathbf{X}_o$ implies that each of these obligations is valid in T as follows. The first and second conjuncts of $\mathbf{X}_o$ imply that the first obligation is valid in T by part 3 of [20, Lemma 14.9]. The first and third conjuncts imply that the second obligation is valid in T by part 5 of [20, Lemma 14.9]. The first and fourth conjuncts imply that the third obligation is valid in T by part 5 of [20, Lemma 14.9]. And the fifth and sixth conjuncts imply, respectively, that the fourth and fifth obligations are valid in T. Therefore, general-MON-to-T is a theory morphism by the Morphism Theorem [20, Theorem 14.16].

We can now transport, as before, the definitions and theorems of MON-1 to D via general-MON-to-T using definition and theorem transportation modules, but we can also transport them using a group transportation module[9]. For example, Thm3 and Def1 can be transported as a group using the following group transportation module:

Group Transportation 5.5 (Transport of Thm3 and Def1 to D)**.**

Name: uniqueness-of-identity-element-and-submonoid-to-D.

Source development: MON-1.

Target development: D.

[9]This kind of module transports a set of definitions and theorems as a group in which order does not matter. A group transportation has nothing to do with the algebraic structure called a group.

Development morphism: general-MON-to-T.

Definitions and theorems:

Thm3: $\forall\, x : M \,.\, (\forall\, y : M \,.\, x \cdot y = y \cdot x = y) \Rightarrow x = \mathsf{e}$
(uniqueness of identity element).

Def1: $\mathsf{submonoid}_{\{M\} \to o} =$
$\lambda\, s : \{M\} \,.\, s \neq \emptyset_{\{M\}} \wedge (\cdot|_{s \times s} \downarrow (s \times s) \to s) \wedge \mathsf{e} \in s$ (submonoid).

Transported definitions and theorems:

Thm3-via-general-MON-to-T:
$\forall\, x : \mathbf{M}_{\{\alpha\}} \,.$
$(\forall\, y : \mathbf{M}_{\{\alpha\}} \,.\, \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(x, y) = \mathbf{F}_{(\alpha\times\alpha)\to\alpha}\,(y, x) = y) \Rightarrow x = \mathbf{E}_\alpha$
(uniqueness of identity element).

Def1-via-general-MON-to-T: $\mathsf{submonoid}_{\{\alpha\} \to o} =$
$\lambda\, s : \mathcal{P}(\mathbf{M}_{\{\alpha\}}) \,.$
$s \neq (\lambda\, x : \mathbf{M}_{\{\alpha\}} \,.\, F_o) \wedge$
$(\mathbf{F}_{(\alpha\times\alpha)\to\alpha}|_{s\times s} \downarrow (s \times s) \to s) \wedge$
$\mathbf{E}_\alpha \in s$ (submonoid).

New target development: D'.

New development morphism: general-MON-1-to-D'.

The abbreviation $\mathcal{P}(\mathbf{M}_{\{\alpha\}})$, which denotes the power set of $\mathbf{M}_{\{\alpha\}}$, is defined in Table 8.

6 Opposite and set monoids

For every monoid $(m, \cdot, e)$, there is (1) an associated monoid $(m, \cdot^{\mathrm{op}}, e)$, where $\cdot^{\mathrm{op}}$ is the opposite of $\cdot$, called the *opposite monoid* of $(m, \cdot, e)$ and (2) a monoid $(\mathcal{P}(m), \odot, \{e\})$, where $\mathcal{P}(m)$ is the power set of m and $\odot$ is the set product on $\mathcal{P}(m)$, called the *set monoid* of $(m, \cdot, e)$.

We will construct a development morphism named MON-to-opposite-monoid from the theory MON to its development MON-1 that maps

$(M, \cdot_{(M\times M)\to M}, \mathsf{e})$

to

$$(M, \cdot^{\mathrm{op}}_{(M \times M) \rightarrow M}, \mathsf{e}).$$

Then we will be able to use this morphism to transport abstract definitions and theorems about monoids to more concrete definitions and theorems about opposite monoids. Here is the definition of MON-to-opposite-monoid:

Development Translation Definition 6.1 (MON to Op. Monoid)**.**

Name: MON-to-opposite-monoid.

Source development: MON.

Target development: MON-1.

Base type mapping:

1. $M \mapsto M$.

Constant mapping:

1. $\cdot_{(M \times M) \rightarrow M} \mapsto \cdot^{\mathrm{op}}_{(M \times M) \rightarrow M}$.
2. $\mathsf{e}_M \mapsto \mathsf{e}_M$.

Since MON-to-opposite-monoid is a normal translation, it has no obligations of the first kind by [20, Lemma 14.10] and two obligations of the second kind which are valid in the top theory of MON-1 by [20, Lemma 14.11]. It has two obligations of the third kind corresponding to the two axioms of MON. These two obligations are logically equivalent to Thm7 and Thm8, respectively, in MON-1, and so these two theorems are obviously valid in the top theory of MON-1. Therefore, MON-to-opposite-monoid is a development morphism from MON to MON-1 by the Morphism Theorem [20, Theorem 14.16].

We can now transport Thm1 via MON-to-opposite-monoid to show that opposite monoids are indeed monoids:

Theorem Transportation 6.2 (Transport of Thm1 to MON-1)**.**

Name: monoid-via-MON-to-opposite-monoid.

Source development: MON.

Target development: MON-1.

Development morphism: MON-to-opposite-monoid.

Theorem:

Thm1: $\mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$
(models of MON define monoids).

Transported theorem:

Thm11 (Thm1-via-MON-to-opposite-monoid):
$\mathsf{MONOID}(U_{\{M\}}, \cdot^{\mathrm{op}}_{(M \times M) \to M}, \mathsf{e}_M)$ (opposite monoids are monoids).

New target development: MON-2.

Similarly, we will construct a development morphism named MON-to-set-monoid from the theory MON to its development MON-2 that maps

$$(M, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$$

to

$$(\{M\}, \odot_{(\{M\} \times \{M\}) \to \{M\}}, \mathsf{E}_{\{M\}}).$$

Then we will be able to use this morphism to transport abstract definitions and theorems about monoids to more concrete definitions and theorems about set monoids. Here is the definition of MON-to-set-monoid:

Development Translation Definition 6.3 (MON to Set Monoid)**.**

Name: MON-to-set-monoid.

Source development: MON.

Target development: MON-2.

Base type mapping:

1. $M \mapsto \{M\}$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \odot_{(\{M\} \times \{M\}) \to \{M\}}$.
2. $\mathsf{e}_M \mapsto \mathsf{E}_{\{M\}}$.

Since MON-to-set-monoid is a normal translation, it has no obligations of the first kind by [20, Lemma 14.10]. It has two obligations of the second kind. The first one is valid in the top theory of MON-2 by part 4 of [20, Lemma 14.9] since $\odot_{(\{M\}\times\{M\})\rightarrow\{M\}}$ beta-reduces by [20, Axiom A4] to a function abstraction which is defined by [20, Axiom A5.11]. The second one is valid in the top theory of MON-2 by part 4 of [20, Lemma 14.9] since $\mathsf{E}_{\{M\}}$ is a function abstraction which is defined by [20, Axiom A5.11]. It has two obligations of the third kind corresponding to the two axioms of MON. These two obligations are Thm9 and Thm10, respectively, in MON-2, and so these two theorems are obviously valid in the top theory of MON-2. Therefore, MON-to-set-monoid is a development morphism from MON to MON-2 by the Morphism Theorem [20, Theorem 14.16].

We can now transport Thm1 via MON-to-set-monoid to show that set monoids are indeed monoids:

Theorem Transportation 6.4 (Transport of Thm1 to MON-2)**.**

Name: monoid-via-MON-to-set-monoid.

Source development: MON.

Target development: MON-2.

Development morphism: MON-to-set-monoid.

Theorem:

Thm1: $\mathsf{MONOID}(U_{\{M\}}, \cdot_{(M\times M)\rightarrow M}, \mathsf{e}_M)$

(models of MON define monoids).

Transported theorem:

Thm12 (Thm1-via-MON-to-set-monoid):
$\mathsf{MONOID}(U_{\{\{M\}\}}, \odot_{(\{M\}\times\{M\})\rightarrow\{M\}}, \mathsf{E}_{\{M\}})$

(set monoids are monoids).

New target development: MON-3.

7 Commutative monoids

A monoid $(m, \cdot, e)$ is *commutative* if $\cdot$ is commutative.

Let $\mathbf{Y}_o$ be the formula

$$\mathsf{COM\text{-}MONOID}(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha\times\alpha)\rightarrow\alpha}, \mathbf{E}_\alpha),$$

where COM-MONOID is the abbreviation introduced by the notational definition given in Table 10. $\mathbf{Y}_o$ asserts that the tuple

$$(\mathbf{M}_{\{\alpha\}}, \mathbf{F}_{(\alpha \times \alpha) \to \alpha}, \mathbf{E}_\alpha)$$

denotes a commutative monoid $(m, \cdot, e)$.

We can define a theory of commutative monoids, named COM-MON, by adding an axiom that says $\cdot$ is commutative to the theory MON using a theory extension module:

Theory Extension 7.1 (Commutative Monoids)**.**

Name: COM-MON.

Extends MON.

New base types:

New constants:

New axioms:

3. $\forall\, x, y : M \,.\, x \cdot y = y \cdot x$ ($\cdot$ is commutative).

Then we can develop the theory COM-MON using the following development definition module:

Development Definition 7.2 (Commutative Monoids 1)**.**

Name: COM-MON-1.

Bottom theory: COM-MON.

Definitions and theorems:

Thm13: COM-MONOID$(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$ (models of COM-MON define commutative monoids).

Def5: $\leq_{M \to M \to o} = \lambda\, x, y : M \,.\, \exists\, z : M \,.\, x \cdot z = y$ (weak order).

Thm14: $\forall\, x : M \,.\, x \leq x$ (reflexivity).

Thm15: $\forall\, x, y, z : M \,.\, (x \leq y \wedge y \leq z) \Rightarrow x \leq z$ (transitivity).

Thm13 states that each model of COM-MON defines a commutative monoid. Def5 defines a weak (nonstrict) order that is a pre-order by Thm14 and Thm15. We could have put Def5, Thm14, and Thm15 in a development of MON since Thm14 and Thm15 do not require that $\cdot$ is commutative, but we have put these in COM-MON instead since $\leq_{M \rightarrow M \rightarrow o}$ is more natural for commutative monoids than for noncommutative monoids.

Since COM-MON is an extension of MON, there is an inclusion (i.e., a theory morphism whose mapping is the identity function) from MON to COM-MON. This inclusion is defined by the following theory translation definition module:

Theory Translation Definition 7.3 (MON to COM-MON)**.**

Name: MON-to-COM-MON.

Source theory: MON.

Target theory: COM-MON.

Base type mapping:

1. $M \mapsto M$.

Constant mapping:

1. $\cdot_{(M \times M) \rightarrow M} \mapsto \cdot_{(M \times M) \rightarrow M}$.
2. $\mathsf{e}_M \mapsto \mathsf{e}_M$.

We will assume that, whenever we define a theory extension T' of a theory T, we also simultaneously define the inclusion from T to T'.

Since MON-to-COM-MON is an inclusion from MON to COM-MON, it is also a development morphism from MON-3 to COM-MON-1 and the definitions and theorems of MON-3 can be freely transported verbatim to COM-MON-1. In the rest of the paper, when a theory T' is an extension of a theory T and D is a development of T, we will assume that the definitions and theorems of D are also definitions and theorems of any trivial or nontrivial development of T' without explicitly transporting them via the inclusion from T to T' as long as there are no name clashes. This assumption is given the name *inclusion transportation convention* in [20, Subsection 14.4.3].

8 Transformation monoids

A very important type of monoid is a monoid composed of transformations of a set. Let s be a nonempty set. Then $(f, \circ, \mathsf{id})$, where f is a set of (partial or total) functions from s to s,

$$\circ : ((s \to s) \times (s \to s)) \to (s \to s)$$

is function composition, and $\mathsf{id} : s \to s$ is the identity function, is a *transformation monoid on* s if f is closed under $\circ$ and $\mathsf{id} \in f$. It is easy to verify that every transformation monoid is a monoid. If f contains every function in the function space $s \to s$, then $(f, \circ, \mathsf{id})$ is clearly a transformation monoid which is called the *full transformation monoid on* s. Let us say that a transformation monoid $(f, \circ, \mathsf{id})$ is *standard* if f contains only total functions. In many developments, nonstandard transformation monoids are ignored, but there is no reason to do that here since Alonzo admits undefined expressions and partial functions.

Consider the following theory ONE-BT of one base type:

Theory Definition 8.1 (One Base Type).

Name: ONE-BT.

Base types: S.

Constants:

Axioms:

We can define the notion of a transformation monoid in a development of this theory, but we must first introduce some general facts about function composition. To do that, we need a theory FUN-COMP with four base types in order to state the associativity theorem for function composition in full generality:

Theory Definition 8.2 (Function Composition).

Name: FUN-COMP.

Base types: A, B, C, D.

Constants:

Axioms:

We introduce two theorems for function composition in a development of FUN-COMP:

Development Definition 8.3 (Function Composition 1)**.**

Name: FUN-COMP-1.

Bottom theory: FUN-COMP.

Definitions and theorems:

Thm16: $\forall\, f : A \to B,\, g : B \to C,\, h : C \to D \,.\, f \circ (g \circ h) = (f \circ g) \circ h$
($\circ$ is associative).

Thm17: $\forall\, f : A \to B \,.\, \mathsf{id}_{A \to A} \circ f = f \circ \mathsf{id}_{B \to B} = f$
(identity functions are left and right identity elements).

The parametric pseudoconstants $\circ_{((\alpha\to\beta)\times(\beta\to\gamma))\to(\alpha\to\gamma)}$ and $\mathsf{id}_{\alpha\to\alpha}$ are defined in Tables 9 and 7, respectively. The infix notation for the application of

$$\circ_{((\alpha\to\beta)\times(\beta\to\gamma))\to(\alpha\to\gamma)}$$

is also defined in Table 9.

Next we define a theory morphism from FUN-COMP to ONE-BT:

Theory Translation Definition 8.4 (FUN-COMP to ONE-BT)**.**

Name: FUN-COMP-to-ONE-BT.

Source theory: FUN-COMP.

Target theory: ONE-BT.

Base type mapping:

1. $A \mapsto S$.
2. $B \mapsto S$.
3. $C \mapsto S$.
4. $D \mapsto S$.

Constant mapping:

The translation FUN-COMP-to-ONE-BT is clearly a theory morphism by the Morphism Theorem [20, Theorem 14.16] since it is a normal translation and FUN-COMP contains no constants or axioms. So we can transport the theorems of FUN-COMP-1 to ONE-BT via FUN-COMP-to-ONE-BT:

Group Transportation 8.5 (Transport of Thm16–Thm17 to ONE-BT)**.**

Name: function-composition-theorems-via-FUN-COMP-to-ONE-BT.

Source development: FUN-COMP-1.

Target development: ONE-BT.

Development morphism: FUN-COMP-to-ONE-BT.

Definitions and theorems:

Thm16: $\forall f : A \rightarrow B,\, g : B \rightarrow C,\, h : C \rightarrow D \,.\, f \circ (g \circ h) = (f \circ g) \circ h$
($\circ$ is associative).

Thm17: $\forall f : A \rightarrow B \,.\, \mathsf{id}_{A \rightarrow A} \circ f = f \circ \mathsf{id}_{B \rightarrow B} = f$
(identity functions are left and right identity elements).

Transported definitions and theorems:

Thm18 (Thm16-via-FUN-COMP-to-ONE-BT):
$\forall f, g, h : S \rightarrow S \,.\, f \circ (g \circ h) = (f \circ g) \circ h$ ($\circ$ is associative).

Thm19 (Thm17-via-FUN-COMP-to-ONE-BT):
$\forall f : S \rightarrow S \,.\, \mathsf{id}_{S \rightarrow S} \circ f = f \circ \mathsf{id}_{S \rightarrow S} = f$
($\mathsf{id}_{S \rightarrow S}$ is an identity element with respect to $\circ$).

New target development: ONE-BT-1.

New development morphism: FUN-COMP-1-to-ONE-BT-1.

We can obtain the theorem that all transformation monoids are monoids almost for free by transporting results from MON-1 to ONE-BT-1. We start by creating the theory morphism from MON to ONE-BT that maps

$$(M, \cdot_{(M \times M) \rightarrow M}, \mathbf{e}_M)$$

to

$$(S \rightarrow S, \circ_{((S \rightarrow S) \times (S \rightarrow S)) \rightarrow (S \rightarrow S)}, \mathsf{id}_{S \rightarrow S}) :$$

Theory Translation Definition 8.6 (MON to ONE-BT)**.**

Name: MON-to-ONE-BT.

Source theory: MON.

Target theory: ONE-BT.

Base type mapping:

1. $M \mapsto S \to S$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \circ_{((S \to S) \times (S \to S)) \to (S \to S)}$.
2. $\mathsf{e}_M \mapsto \mathsf{id}_{S \to S}$.

The theory translation MON-to-ONE-BT is normal so that it has no obligations of the first kind by [20, Lemma 14.10]. It has two obligations of the second kind. These are valid in ONE-BT by part 4 of [20, Lemma 14.9] since $\circ_{((S \to S) \times (S \to S)) \to (S \to S)}$ and $\mathsf{id}_{S \to S}$ are function abstractions which are defined by [20, Axiom A5.11]. It has two obligations of the third kind corresponding to the two axioms of MON. The two obligations are Thm18 and Thm19, respectively, in ONE-BT-1, and so these two theorems are obviously valid in the top theory of ONE-BT-1. Therefore, MON-to-ONE-BT is a theory morphism from MON to ONE-BT by the Morphism Theorem [20, Theorem 14.16].

We can transport Def1, the definition of $\mathsf{submonoid}_{\{M\} \to o}$, and Thm4, the theorem that says all submonoids are monoids, to ONE-BT-1 via MON-to-ONE-BT by a group transportation module:

Group Transportation 8.7 (Transport of Def1 & Thm2 to ONE-BT-1)**.**

Name: submonoids-via-MON-to-ONE-BT.

Source development: MON-1.

Target development: ONE-BT-1.

Development morphism: MON-to-ONE-BT.

Definitions and theorems:

Def1: $\mathsf{submonoid}_{\{M\} \to o} =$
$\lambda\, s : \{M\} \,.\, s \neq \emptyset_{\{M\}} \wedge (\cdot|_{s \times s} \downarrow (s \times s) \to s) \wedge \mathsf{e} \in s$ (submonoid).

Thm4: $\forall\, s : \{M\} \,.\, \mathsf{submonoid}\, s \Rightarrow \mathsf{MONOID}(s, \cdot|_{s \times s}, \mathsf{e})$
(submonoids are monoids).

Transported definitions and theorems:

Def6 (Def1-via-MON-to-ONE-BT): $\mathsf{trans\text{-}monoid}_{\{S \to S\} \to o} =$
$\lambda\, s : \{S \to S\}\ .$
$s \neq \emptyset_{\{S \to S\}} \wedge$
$(\circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{s \times s} \downarrow (s \times s) \to s) \wedge$
$\mathsf{id}_{S \to S} \in s$ (transformation monoid).

Thm20 (Thm4-via-MON-to-ONE-BT):
$\forall\, s : \{S \to S\}\ .$
$\mathsf{trans\text{-}monoid}\, s \Rightarrow \mathsf{MONOID}(s, \circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{s \times s}, \mathsf{id}_{S \to S})$
(transformation monoids are monoids).

New target development: ONE-BT-2.

New development morphism: MON-1-to-ONE-BT-2.

trans-monoid is a predicate that is true when it is applied to a set of functions of $S \to S$ that forms a transformation monoid. Thm20 says that every transformation monoid — including the full transformation monoid — is a monoid.

9 Monoid actions

A *(left) monoid action* is a mathematical structure $(m, s, \cdot, e, \mathsf{act})$ where $(m, \cdot, e)$ is a monoid and $\mathsf{act} : (m \times s) \to s$ is a function such that

(1) $x\ \mathsf{act}\ (y\ \mathsf{act}\ z) = (x \cdot y)\ \mathsf{act}\ z$

for all $x, y \in m$ and $z \in s$ and

(2) $e\ \mathsf{act}\ z = z$

for all $z \in s$. We say in this case that the monoid $(m, \cdot, e)$ *acts on* the set s *by* the function act.

Let $\mathbf{Z}_o$ be the formula

$$\mathsf{MON\text{-}ACTION}(\mathbf{M}_{\{\alpha\}}, \mathbf{S}_{\{\beta\}}, \mathbf{F}_{(\alpha \times \alpha) \to \alpha}, \mathbf{E}_\alpha, \mathbf{G}_{(\alpha \times \beta) \to \beta}),$$

where MON-ACTION is the abbreviation introduced by the notational definition given in Table 10. $\mathbf{Z}_o$ asserts that the tuple

$$(\mathbf{M}_{\{\alpha\}}, \mathbf{S}_{\{\beta\}}, \mathbf{F}_{(\alpha \times \alpha) \to \alpha}, \mathbf{E}_\alpha, \mathbf{G}_{(\alpha \times \beta) \to \beta})$$

denotes a monoid action $(m, s, \cdot, e, \mathsf{act})$.

A theory of monoid actions is defined as an extension of the theory of monoids:

Theory Extension 9.1 (Monoid Actions)**.**

Name: MON-ACT.

Extends MON.

New base types: S.

New constants: $\mathsf{act}_{(M \times S) \to S}$.

New axioms:

3. $\forall\, x, y : M,\ s : S\ .\ x\ \mathsf{act}\ (y\ \mathsf{act}\ s) = (x \cdot y)\ \mathsf{act}\ s$ (act is compatible with $\cdot$).

4. $\forall\, s : S\ .\ \mathsf{e}\ \mathsf{act}\ s = s$ (act is compatible with e).

We begin a development of MON-ACT by adding the definitions and theorems below:

Development Definition 9.2 (Monoid Actions 1)**.**

Name: MON-ACT-1.

Bottom theory: MON-ACT.

Definitions and theorems:

Thm21: $\mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{S\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S})$ (models of MON-ACT define monoid actions).

Thm22: $\mathsf{TOTAL}(\mathsf{act}_{(M \times S) \to S})$ (act is total).

Def7: $\mathsf{orbit}_{S \to \{S\}} = \lambda\, s : S\ .\ \{t : S \mid \exists\, x : M\ .\ x\ \mathsf{act}\ s = t\}$ (orbit).

Def8: $\mathsf{stabilizer}_{S \to \{M\}} = \lambda\, s : S\ .\ \{x : M \mid x\ \mathsf{act}\ s = s\}$ (stabilizer).

Thm23: $\forall\, s : S\ .\ \mathsf{submonoid}\ (\mathsf{stabilizer}\ s)$ (stabilizers are submonoids).

Thm21 states that each model of MON-ACTION defines a monoid action. Thm22 says that $\mathsf{act}_{(M \times S) \to S}$ is total (which is implied by the third axiom of MON-ACTION). Def7 and Def8 introduce the concepts of an orbit and a stabilizer. And Thm23 states that a stabilizer of a monoid action $(m, s, \cdot, e, \mathsf{act})$ is a submonoid of the monoid $(m, \cdot, e)$. The power of this machinery — monoid actions with orbits and stabilizers — is low with arbitrary monoids but very high with groups, i.e., monoids in which every element has an inverse.

Monoid actions are common in monoid theory. We will present two important examples of monoid actions. The first is the monoid action $(m, m, \cdot, e, \cdot)$ such that the monoid $(m, \cdot, e)$ acts on the set m of its elements by its function $\cdot$. We formalize this by creating the theory morphism from MON-ACT to MON that maps

$$(M, S, \cdot_{(M \times M) \to M}, \mathbf{e}_M, \mathsf{act}_{(M \times S) \to S})$$

to

$$(M, M, \cdot_{(M \times M) \to M}, \mathbf{e}_M, \cdot_{(M \times M) \to M}):$$

Theory Translation Definition 9.3 (MON-ACT to MON)**.**

Name: MON-ACT-to-MON.

Source theory: MON-ACT.

Target theory: MON.

Base type mapping:

1. $M \mapsto M$.
2. $S \mapsto M$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \cdot_{(M \times M) \to M}$.
2. $\mathsf{e}_M \mapsto \mathsf{e}_M$.
3. $\mathsf{act}_{(M \times S) \to S} \mapsto \cdot_{(M \times M) \to M}$.

It is an easy exercise to verify, arguing as we have above, that MON-ACT-to-MON is a theory morphism.

We can now transport Thm21 from MON-ACT to MON-3 via MON-ACT-to-MON to show that the action of a monoid $(m, \cdot, e)$ on m by $\cdot$ is a monoid action:

Theorem Transportation 9.4 (Transport of Thm21 to MON-3)**.**

Name: monoid-action-via-MON-ACT-to-MON.

Source development: MON-ACT.

Target development: MON-3.

Development morphism: MON-ACT-to-MON.

Theorem:

Thm21: $\mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{S\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S})$
(models of MON-ACT define monoid actions).

Transported theorem:

Thm24 (Thm21-via-MON-ACT-to-MON):
$\mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \cdot_{(M \times M) \to M})$
(first example is a monoid action).

New target development: MON-4.

The second example is a standard transformation monoid $(f, \circ, \mathsf{id})$ on s acting on s by the function that applies a transformation to a member of s. (Note that all the functions in f are total by virtue of the transformation monoid being standard.) We formalize this example as a theory morphism from MON-ACT to ONE-BT extended with a set constant that denotes a standard transformation monoid. Here is the extension with a set constant $\mathsf{F}_{\{S \to S\}}$ and two axioms:

Theory Extension 9.5 (One Base Type with a Set Constant)**.**

Name: ONE-BT-with-SC.

Extends ONE-BT.

New base types:

New constants: $\mathsf{F}_{\{S \to S\}}$.

New axioms:

1. $\mathsf{trans\text{-}monoid}\,\mathsf{F}$ (F forms a transformation monoid).
2. $\forall\, f : \mathsf{F} \,.\, \mathsf{TOTAL}(f)$ (the members of F are total functions).

And here is the theory morphism from MON-ACT to ONE-BT-with-SC that maps

$$(M, S, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S})$$

to

$$(\mathsf{F}_{\{S \to S\}}, S, \circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{\mathsf{F} \times \mathsf{F}}, \mathsf{id}_{S \to S}, \bullet_{((S \to S) \times S) \to S}|_{\mathsf{F} \times S}) :$$

Theory Translation Definition 9.6 (MON-ACT to ONE-BT-with-SC)**.**

Name: MON-ACT-to-ONE-BT-with-SC.

Source theory: MON-ACT.

Target theory: ONE-BT-with-SC.

Base type mapping:

1. $M \mapsto \mathsf{F}_{\{S \to S\}}$.

2. $S \mapsto S$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{\mathsf{F} \times \mathsf{F}}$.

2. $\mathsf{e}_M \mapsto \mathsf{id}_{S \to S}$.

3. $\mathsf{act}_{(M \times S) \to S} \mapsto \bullet_{((S \to S) \times S) \to S}|_{\mathsf{F} \times S}$.

The parametric pseudoconstant $\bullet_{((S \to S) \times S) \to S}|_{\mathsf{F} \times S}$ is defined in Table 9. It is a straightforward exercise to verify, arguing as we have above, that MON-ACT-to-ONE-BT-with-SC is a theory morphism.

We can now transport Thm21 from MON-ACT to ONE-BT-with-S via MON-ACT-to-ONE-BT-with-SC to show that a standard transformation monoid $(f, \circ, \mathsf{id})$ on s acting on s by the function that applies a (total) transformation to a member of s is a monoid action:

Theorem Transportation 9.7 (Trans. of Thm21 to ONE-BT-with-SC)**.**

Name: monoid-action-via-MON-ACT-to-ONE-BT-with-SC.

Source development: MON-ACT.

Target development: ONE-BT-with-SC.

Development morphism: MON-ACT-to-ONE-BT-with-SC.

Theorem:

Thm21: $\mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{S\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S})$
(models of MON-ACT define monoid actions).

Transported theorem:

Thm25 (Thm21-via-MON-ACT-to-ONE-BT-with-SC):
$\mathsf{MON\text{-}ACTION}(\mathsf{F}_{\{S \to S\}},$
$U_{\{S\}},$
$\circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{\mathsf{F} \times \mathsf{F}},$
$\mathsf{id}_{S \to S},$
$\bullet_{((S \to S) \times S) \to S}|_{\mathsf{F} \times S})$
(second example is a monoid action).

New target development: ONE-BT-with-SC-1.

10 Monoid homomorphisms

Roughly speaking, a *monoid homomorphism* is a structure-preserving mapping from one monoid to another.

Let $\mathbf{W}_o$ be the formula

$$\mathsf{MON\text{-}HOMOM}(\mathbf{M}^1_{\{\alpha\}}, \mathbf{M}^2_{\{\beta\}}, \mathbf{F}^1_{(\alpha \times \alpha) \to \alpha}, \mathbf{E}^1_\alpha, \mathbf{F}^2_{(\beta \times \beta) \to \beta}, \mathbf{E}^2_\beta, \mathbf{H}_{\alpha \to \beta}),$$

where MON-HOMOM is the abbreviation introduced by the notational definition given in Table 10. $\mathbf{W}_o$ asserts that the tuple

$$(\mathbf{M}^1_{\{\alpha\}}, \mathbf{M}^2_{\{\beta\}}, \mathbf{F}^1_{(\alpha \times \alpha) \to \alpha}, \mathbf{E}^1_\alpha, \mathbf{F}^2_{(\beta \times \beta) \to \beta}, \mathbf{E}^2_\beta, \mathbf{H}_{\alpha \to \beta})$$

denotes a mathematical structure $(m_1, m_2, \cdot_1, e_1, \cdot_2, e_2, h)$ where $(m_1, \cdot_1, e_1)$ is a monoid, $(m_2, \cdot_2, e_2)$ is a monoid, and $h : m_1 \to m_2$ is a monoid homomorphism from $(m_1, \cdot_1, e_1)$ to $(m_2, \cdot_2, e_2)$.

The notion of a monoid homomorphism is captured in the theory MON-HOM:

Theory Definition 10.1 (Monoid Homomorphisms).

Name: MON-HOM.

Base types: M_1, M_2.

Constants: $\cdot_{(M_1 \times M_1) \to M_1}$, e_{M_1}, $\cdot_{(M_2 \times M_2) \to M_2}$, e_{M_2}, $\mathsf{h}_{M_1 \to M_2}$.

Axioms:

1. $\forall x, y, z : M_1 \,.\, x \cdot (y \cdot z) = (x \cdot y) \cdot z$ ($\cdot_{(M_1 \times M_1) \to M_1}$ is associative).
2. $\forall x : M_1 \,.\, \mathsf{e} \cdot x = x \cdot \mathsf{e} = x$ (e_{M_1} is an identity element).
3. $\forall x, y, z : M_2 \,.\, x \cdot (y \cdot z) = (x \cdot y) \cdot z$ ($\cdot_{(M_2 \times M_2) \to M_2}$ is associative).

4. $\forall\, x : M_2 \,.\, \mathsf{e} \cdot x = x \cdot \mathsf{e} = x$ (e_{M_2} is an identity element).

5. $\forall\, x, y : M_1 \,.\, \mathsf{h}\,(x \cdot y) = (\mathsf{h}\,x) \cdot (\mathsf{h}\,y)$ (first homomorphism property).

6. $\mathsf{h}\,\mathsf{e}_{M_1} = \mathsf{e}_{M_2}$ (second homomorphism property).

$\mathsf{h}_{M_1 \rightarrow M_2}$ denotes a monoid homomorphism from the monoid denoted by

$$(M_1, \cdot_{(M_1 \times M_1) \rightarrow M_1}, \mathsf{e}_{M_1})$$

to the monoid denoted by

$$(M_2, \cdot_{(M_2 \times M_2) \rightarrow M_2}, \mathsf{e}_{M_2}).$$

Here is a simple development of MON-HOM:

Development Definition 10.2 (Monoid Homomorphisms 1)**.**

Name: MON-HOM-1.

Bottom theory: MON-HOM.

Definitions and theorems:

Thm26:
$\mathsf{MON\text{-}HOM}(U_{\{M_1\}}, U_{\{M_2\}}, \cdot_{(M_1 \times M_1) \rightarrow M_1}, \mathsf{e}_{M_1}, \cdot_{(M_2 \times M_2) \rightarrow M_2}, \mathsf{e}_{M_2}, \mathsf{h}_{M_1 \rightarrow M_2})$
(models of MON-HOM define monoid homomorphisms).

Thm27: $\mathsf{TOTAL}(\mathsf{h}_{M_1 \rightarrow M_2})$ ($\mathsf{h}_{M_1 \rightarrow M_2}$ is total).

There are embeddings (i.e., theory morphisms whose mappings are injective) from MON to the two copies of MON within MON-HOM defined by the following two theory translation definitions:

Theory Translation Definition 10.3 (First MON to MON-HOM)**.**

Name: first-MON-to-MON-HOM.

Source theory: MON.

Target theory: MON-HOM.

Base type mapping:

1. $M \mapsto M_1$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \cdot_{(M_1 \times M_1) \to M_1}$.
2. $\mathsf{e}_M \mapsto \mathsf{e}_{M_1}$.

Theory Translation Definition 10.4 (Second MON to MON-HOM)**.**

Name: second-MON-to-MON-HOM.

Source theory: MON.

Target theory: MON-HOM.

Base type mapping:

1. $M \mapsto M_2$.

Constant mapping:

1. $\cdot_{(M \times M) \to M} \mapsto \cdot_{(M_2 \times M_2) \to M_2}$.
2. $\mathsf{e}_M \mapsto \mathsf{e}_{M_2}$.

An example of a monoid homomorphism from the monoid denoted by

$$(M, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$$

to the monoid denoted by

$$(\{M\}, \odot_{(\{M\} \times \{M\}) \to \{M\}}, \mathsf{E}_{\{M\}})$$

is the function that maps a member x of the denotation of M to the singleton $\{x\}$. This monoid homomorphism is formalized by the following development morphism:

Development Translation Definition 10.5 (MON-HOM to MON)**.**

Name: MON-HOM-to-MON-4.

Source development: MON-HOM.

Target development: MON-4.

Base type mapping:

1. $M_1 \mapsto M$.

2. $M_2 \mapsto \{M\}$.

Constant mapping:

1. $\cdot_{(M_1 \times M_1) \to M_1} \mapsto \cdot_{(M \times M) \to M}$.

2. $\mathsf{e}_{M_1} \mapsto \mathsf{e}_M$.

3. $\cdot_{(M_2 \times M_2) \to M_2} \mapsto \odot_{(\{M\} \times \{M\}) \to \{M\}}$.

4. $\mathsf{e}_{M_2} \mapsto \mathsf{E}_{\{M\}}$.

5. $\mathsf{h}_{M_1 \to M_2} \mapsto \lambda\, x : M \,.\, \{x\}$.

It is a straightforward exercise to verify that HOM-MON-to-MON-4 is a theory morphism by the arguments we employed above.

We can now transport Thm26 from MON-HOM to MON-4 via MON-HOM-to-MON-4 to show the example is a monoid homomorphism:

Theorem Transportation 10.6 (Transport of Thm26 to MON-4)**.**

Name: monoid-action-via-MON-HOM-to-MON-4.

Source development: MON-HOM.

Target development: MON-4.

Development morphism: MON-HOM-to-MON-4.

Theorem:

Thm26:
$$\mathsf{MON\text{-}HOM}(U_{\{M_1\}}, U_{\{M_2\}}, \cdot_{(M_1 \times M_1) \to M_1}, \mathsf{e}_{M_1}, \cdot_{(M_2 \times M_2) \to M_2}, \mathsf{e}_{M_2}, \mathsf{h}_{M_1 \to M_2})$$

(models of MON-HOM define monoid homomorphisms).

Transported theorem:

Thm28 (Thm26-via-MON-HOM-to-MON-4)
MON-HOM($U_{\{M\}}$,
$U_{\{\{M\}\}}$,
$\cdot_{(M \times M) \to M}$,
e_M,
$\odot_{(\{M\} \times \{M\}) \to \{M\}}$,
$\mathsf{E}_{\{M\}}$,
$\lambda\, x : M \,.\, \{x\}$) (example is a monoid homomorphism).

New target development: MON-5.

11 Monoids over real number arithmetic

We need machinery concerning real number arithmetic to express some concepts about monoids. For instance, an iterated product operator for monoids involves integers. To formalize these kinds of concepts, we need a theory of monoids that includes real number arithmetic. Chapter 13 of [20] presents COF, a theory of complete ordered fields. COF is categorical in the standard sense (see [20]). That is, it has a single standard model up to isomorphism that defines the structure of real number arithmetic.

We define a theory of monoids over COF by extending COF with the language and axioms of MON:

Theory Extension 11.1 (Monoids over COF)**.**

Name: MON-over-COF.

Extends COF.

New base types: M.

New constants: $\cdot_{(M \times M) \to M}$, e_M.

New axioms:

19. $\forall\, x, y, z : M \,.\, x \cdot (y \cdot z) = (x \cdot y) \cdot z$ ($\cdot$ is associative).

20. $\forall\, x : M \,.\, \mathsf{e} \cdot x = x \cdot \mathsf{e} = x$ (e is an identity element).

We can now define an iterated product operator for monoids in a development of MON-over-COF-1:

$\left(\prod_{\mathbf{i}=\mathbf{M}_R}^{\mathbf{N}_R} \mathbf{A}_M\right)$	stands for	$\mathsf{prod}_{R \rightarrow R \rightarrow (R \rightarrow M) \rightarrow M}\ \mathbf{M}_R\, \mathbf{N}_R\, (\lambda\, \mathbf{i} : R \,.\, \mathbf{A}_M)$.

Table 11: Notational Definition for Monoids: Iterated Product Operator

Development Definition 11.2 (Monoids over COF 1)**.**

Name: MON-over-COF-1.

Bottom theory: MON-over-COF.

Definitions and theorems:

Def9: $\mathsf{prod}_{R \rightarrow R \rightarrow (R \rightarrow M) \rightarrow M} =$
$\mathrm{I}\, f : Z_{\{R\}} \rightarrow Z_{\{R\}} \rightarrow (Z_{\{R\}} \rightarrow M) \rightarrow M \,.$
$\forall\, m, n : Z_{\{R\}},\, g : Z_{\{R\}} \rightarrow M \,.\, f\, m\, n\, g \simeq$
$(m > n \mapsto \mathsf{e} \mid (f\, m\, (n-1)\, g) \cdot (g\, n))$ (iterated product).

Thm29: $\forall\, m : Z_{\{R\}},\, g : Z_{\{R\}} \rightarrow M \,.\, \left(\prod_{i=m}^{m} g\, i\right) \simeq g\, m$
(trivial product).

Thm30: $\forall\, m, k, n : Z_{\{R\}},\, g : Z_{\{R\}} \rightarrow M \,.$
$m < k < n \Rightarrow \left(\prod_{i=m}^{k} g\, i\right) \cdot \left(\prod_{i=k+1}^{n} g\, i\right) \simeq \prod_{i=m}^{n} g\, i$
(extended iterated product).

We are utilizing the notation for the iterated product operator defined in Table 11. $Z_{\{R\}}$ is a quasitype defined in the development COF-dev-2 of COF found in [20] that denotes the set of integers. ($Z_{\{R\}}$ is automatically available in MON-over-COF by the inclusion transportation convention presented in Section 7.) Def9 defines the iterated product operator, and Thm29 and Thm30 are two theorems about the operator.

We can similarly define extensions of MON over COF. For example, here is a theory of commutative monoids over COF and a development of it:

Theory Extension 11.3 (Commutative Monoids over COF)**.**

Name: COM-MON-over-COF.

Extends MON-over-COF.

New base types:

New constants:

New axioms:

21. $\forall\, x, y : M \,.\, x \cdot y = y \cdot x$ ($\cdot$ is commutative).

Development Definition 11.4 (Com. Monoids over COF 1)**.**

Name: COM-MON-over-COF-1.

Bottom theory: COM-MON-over-COF.

Definitions and theorems:

Thm31: $\forall\, m, n : Z_{\{R\}},\ g, h : Z_{\{R\}} \to M \,.$
$$\left(\prod_{i=m}^{n} g\, i\right) \cdot \left(\prod_{i=m}^{n} h\, i\right) \simeq \prod_{i=m}^{n} (g\, i) \cdot (h\, i)$$
(product of iterated products).

Notice that this theorem holds only if $\cdot$ is commutative.

For another example, here is a theory of commutative monoid actions over COF and a development of it:

Theory Extension 11.5 (Commutative Monoid Actions over COF)**.**

Name: COM-MON-ACT-over-COF.

Extends COM-MON-over-COF.

New base types: S.

New constants: $\mathsf{act}_{(M \times S) \to S}$.

New axioms:

22. $\forall\, x, y : M,\ s : S \,.\, x\ \mathsf{act}\ (y\ \mathsf{act}\ s) = (x \cdot y)\ \mathsf{act}\ s$ (act is compatible with $\cdot$).

23. $\forall\, s : S \,.\, \mathsf{e}\ \mathsf{act}\ s = s$ (act is compatible with e).

Development Definition 11.6 (Com. Monoid Actions over COF 1)**.**

Name: COM-MON-ACT-over-COF-1.

Bottom theory: COM-MON-ACT-over-COF.

Definitions and theorems:

Thm32: $\forall\, x, y : M,\ s : S \,.\, x\ \mathsf{act}\ (y\ \mathsf{act}\ s) = y\ \mathsf{act}\ (x\ \mathsf{act}\ s)$
(act has commutative-like property).

$\mathsf{sequences}_{\{\alpha\to\beta\}}$	stands for	$\mathbf{C}^{N}_{\{\alpha\}} \to \beta$.
$\langle\!\langle\beta\rangle\!\rangle$	stands for	$\mathsf{sequences}_{\{\alpha\to\beta\}}$.
$\mathsf{streams}_{\{\alpha\to\beta\}}$	stands for	$\{s : \langle\!\langle\beta\rangle\!\rangle \mid \mathsf{TOTAL}(s)\}$.
$\langle\beta\rangle$	stands for	$\mathsf{streams}_{\{\alpha\to\beta\}}$.
$\mathsf{lists}_{\{\alpha\to\beta\}}$	stands for	$\{s : \langle\!\langle\beta\rangle\!\rangle \mid \exists\, n : \mathbf{C}^{N}_{\{\alpha\}} \,.\, \forall\, m : \mathbf{C}^{N}_{\{\alpha\}} \,.\, (s\,m)\!\downarrow\, \Leftrightarrow \mathbf{C}^{\le}_{\alpha\to\alpha\to o}\, m\,(\mathbf{C}^{P}_{\alpha\to\alpha}\, n)\}$.
$[\beta]$	stands for	$\mathsf{lists}_{\{\alpha\to\beta\}}$.
$\mathsf{cons}_{\beta\to(\alpha\to\beta)\to(\alpha\to\beta)}$	stands for	$\lambda\, x : \beta \,.\, \lambda\, s : \langle\!\langle\beta\rangle\!\rangle \,.\, \lambda\, n : \mathbf{C}^{N}_{\{\alpha\}} \,.\, n = \mathbf{C}^{0}_{\alpha} \mapsto x \mid s\,(\mathbf{C}^{P}_{\alpha\to\alpha}\, n)$.
$(\mathbf{A}_\beta :: \mathbf{B}_{\alpha\to\beta})$	stands for	$\mathsf{cons}_{\beta\to(\alpha\to\beta)\to(\alpha\to\beta)}\ \mathbf{A}_\beta\, \mathbf{B}_{\alpha\to\beta}$.
$\mathsf{nil}_{\alpha\to\beta}$	stands for	$\Delta_{\alpha\to\beta}$.
$[\,]_{\alpha\to\beta}$	stands for	$\mathsf{nil}_{\alpha\to\beta}$.
$[\mathbf{A}_\beta]$	stands for	$(\mathbf{A}_\beta :: [\,]_{\alpha\to\beta})$.
$[\mathbf{A}^{1}_{\beta}, \ldots, \mathbf{A}^{n}_{\beta}]$	stands for	$(\mathbf{A}^{1}_{\beta} :: [\mathbf{A}^{2}_{\beta}, \ldots, \mathbf{A}^{n}_{\beta}])$ where $n \ge 2$.
$\mathsf{len}_{(\alpha\to\beta)\to\alpha}$	stands for	$\mathrm{I}\, f : [\beta] \to \mathbf{C}^{N}_{\{\alpha\}} \,.\, f\,[\,]_{\alpha\to\beta} = \mathbf{C}^{0}_{\alpha} \wedge \forall\, x : \beta,\, s : [\beta] \,.\, f\,(x :: s) = \mathbf{C}^{+}_{\alpha\to\alpha\to\alpha}\,(f\,s)\,(\mathbf{C}^{S}_{\alpha\to\alpha}\, \mathbf{C}^{0}_{\alpha})$.
$\lvert\mathbf{A}_{\alpha\to\beta}\rvert$	stands for	$\mathsf{len}_{(\alpha\to\beta)\to\alpha}\ \mathbf{A}_{\alpha\to\beta}$.
$++_{(\alpha\to\beta)\to(\alpha\to\beta)\to(\alpha\to\beta)}$	stands for	$\mathrm{I}\, f : [\beta] \to [\beta] \to [\beta] \,.\, \forall\, t : [\beta] \,.\, f\,[\,]_{\alpha\to\beta}\, t = t \wedge \forall\, x : \beta,\, s, t : [\beta] \,.\, f\,(x :: s)\,t = (x :: f\,s\,t)$.

Table 12: Notational Definitions for Sequences

12 Monoid theory applied to strings

In this section we will show how the machinery of our monoid theory formalization can be applied to a theory of strings over an abstract alphabet. A string over an alphabet A is a finite sequence of values from A. The finite sequence s can be represented as a partial function $s : \mathbb{N} \to A$ such that, for some $n \in \mathbb{N}$, $s(m)$ is defined iff $m < n$.

In Table 12 we introduce compact notation for finite (and infinite) sequences represented in this manner. The notation requires a system of natural numbers as defined in Chapter 11 of [20]. We also introduce some special notation for strings in Table 13.

The development $\mathsf{COF\text{-}dev\text{-}2}$ of the theory COF presented in Chapter 13 of [20] includes a system of natural numbers [20, Proposition 13.11]. Therefore, we can define a theory of strings as an extension of COF plus a base type A that represents an abstract alphabet:

$(\mathbf{X}_{R\to A}\mathbf{Y}_{R\to A})$	stands for	$\mathbf{X}_{R\to A}$ cat $\mathbf{Y}_{R\to A}$.
$(\mathbf{S}_{\{R\to A\}}\mathbf{T}_{\{R\to A\}})$	stands for	$\mathbf{S}_{\{R\to A\}}$ set-cat $\mathbf{T}_{\{R\to A\}}$.
$\left(\overset{\mathbf{N}_R}{\underset{\mathbf{i}=\mathbf{M}_R}{\mathsf{cat}}}\ \mathbf{A}_{R\to A}\right)$	stands for	$\mathsf{iter\text{-}cat}_{R\to R\to(R\to(R\to A))\to(R\to A)}\ \mathbf{M}_R\,\mathbf{N}_R\,(\lambda\,\mathbf{i}:R\,.\,\mathbf{A}_{R\to A})$.

Table 13: Notational Definitions for Monoids: Special Notation

Theory Extension 12.1 (Strings).

Name: STR.

Extends COF.

New base types: A.

New constants:

New axioms:

Since STR is an extension of COF, we can assume that STR-1 is a development of STR that contains the 7 definitions of COF-dev-2 named as COF-Def1, ..., COF-Def7 and the 22 theorems of COF-dev-2 named as COF-Thm1, ..., COF-Thm22. We can extend STR-1 as follows to include the basic definitions and theorems of strings:

Development Extension 12.2 (Strings 2).

Name: STR-2.

Extends STR-1.

New definitions and theorems:

Def10: $\mathsf{str}_{\{R\to A\}} = [A]$ (string quasitype).

Def11: $\epsilon_{R\to A} = [\,]_{R\to A}$ (empty string).

Def12: $\mathsf{cat}_{((R\to A)\times(R\to A))\to(R\to A)} = {++}_{(R\to A)\to(R\to A)\to(R\to A)}$ (concatenation).

Thm33: $\forall\, x : \mathsf{str}\,.\,\epsilon x = x\epsilon = x$ (ϵ is an identity element).

Thm34: $\forall\, x, y, z : \mathsf{str}\,.\,x(yz) = (xy)z$ (cat is associative).

Def10–Def12 utilize the compact notation introduced in Table 12 and Thm33–Thm34 utilize the compact notation introduced in Table 13.

We can define a development translation from MON-over-COF to STR-2 as follows:

Development Translation Definition 12.3 (MON-over-COF to STR-2)**.**

Name: MON-over-COF-to-STR-2.

Source development: MON-over-COF.

Target development: STR-2.

Base type mapping:

1. $R \mapsto R$.

2. $M \mapsto \mathsf{str}_{\{R \rightarrow A\}}$.

Constant mapping:

1. $0_R \mapsto 0_R$.

$\vdots$

10. $\mathsf{lub}_{R \rightarrow \{R\} \rightarrow o} \mapsto \mathsf{lub}_{R \rightarrow \{R\} \rightarrow o}$.

11. $\cdot_{(M \times M) \rightarrow M} \mapsto \mathsf{cat}_{((R \rightarrow A) \times (R \rightarrow A)) \rightarrow (R \rightarrow A)}$.

12. $\mathsf{e}_M \mapsto \epsilon_{R \rightarrow A}$.

MON-over-COF-to-STR-2 has one obligation of the first kind for the mapped base type M, which is clearly valid since $\mathsf{str}_{\{R \rightarrow A\}}$ is nonempty. MON-over-COF-to-STR-2 has 12 obligations of the second kind for the 12 mapped constants. The first 10 are trivially valid. The last 2 are valid by Def12 and Def11, respectively. And MON-over-COF-to-STR-2 has 20 obligations of the third kind for the 20 axioms of MON-over-COF. The first 18 are trivially valid. The last 2 are valid by Thm34 and Thm33, respectively. Therefore, MON-over-COF-to-STR-2 is a development morphism from the theory MON-over-COF to the development STR-2 by the Morphism Theorem [20, Theorem 14.16].

The development morphism MON-over-COF-to-STR-2 allows us to transport definitions and theorems about monoids to the development STR-2. Here are five examples transported as a group:

Group Transportation 12.4 (Transport to STR-2)**.**

Name: monoid-machinery-via-MON-over-COF-1-to-STR-2.

Source development: MON-over-COF-1.

Target development: STR-2.

Development morphism: MON-over-COF-to-STR-2.

Definitions and theorems:

Thm1: $\mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \rightarrow M}, \mathsf{e}_M)$

(models of MON define monoids).

Def3: $\odot_{(\{M\} \times \{M\}) \rightarrow \{M\}} = \mathsf{set\text{-}op}_{((M \times M) \rightarrow M) \rightarrow ((\{M\} \times \{M\}) \rightarrow \{M\})} \cdot$

(set product).

Def4: $\mathsf{E}_{\{M\}} = \{\mathsf{e}_M\}$ (set identity element).

Thm12 (Thm1-via-MON-to-set-monoid):
$\mathsf{MONOID}(U_{\{\{M\}\}}, \odot_{(\{M\} \times \{M\}) \rightarrow \{M\}}, \mathsf{E}_{\{M\}})$

(set monoids are monoids).

Def9: $\mathsf{prod}_{R \rightarrow R \rightarrow (R \rightarrow M) \rightarrow M} =$
$\mathrm{I}\, f : Z_{\{R\}} \rightarrow Z_{\{R\}} \rightarrow (Z_{\{R\}} \rightarrow M) \rightarrow M\ .$
$\forall\, m, n : Z_{\{R\}},\, g : Z_{\{R\}} \rightarrow M\ .\ f\, m\, n\, g \simeq$
$(m > n \mapsto \mathsf{e} \mid (f\, m\, (n-1)\, g) \cdot (g\, n))$ (iterated product).

Transported definitions and theorems:

Thm35 (Thm1-via-MON-over-COF-to-STR-2):
$\mathsf{MONOID}(\mathsf{str}_{\{R \rightarrow A\}}, \mathsf{cat}_{((R \rightarrow A) \times (R \rightarrow A)) \rightarrow (R \rightarrow A)}, \epsilon_{R \rightarrow A})$

(strings form a monoid).

Def13 (Def3-via-MON-over-COF-to-STR-2):
$\mathsf{set\text{-}cat}_{(\{R \rightarrow A\} \times \{R \rightarrow A\}) \rightarrow \{R \rightarrow A\}} =$
$\mathsf{set\text{-}op}_{(((R \rightarrow A) \times (R \rightarrow A)) \rightarrow (R \rightarrow A)) \rightarrow ((\{R \rightarrow A\} \times \{R \rightarrow A\}) \rightarrow \{R \rightarrow A\})}\ \mathsf{cat}$

(set concatenation).

Def14 (Def4-via-MON-over-COF-to-STR-2):
$\mathsf{E}_{\{R \rightarrow A\}} = \{\epsilon_{R \rightarrow A}\}$ (set identity element).

Thm36 (Thm12-via-MON-over-COF-1-to-STR-2):
$\mathsf{MONOID}(\mathcal{P}(\mathsf{str}_{\{R \rightarrow A\}}), \mathsf{set\text{-}cat}_{(\{R \rightarrow A\} \times \{R \rightarrow A\}) \rightarrow \{R \rightarrow A\}}, \mathsf{E}_{\{R \rightarrow A\}})$

(string sets form a monoid).

Def15 (Def9-via-MON-over-COF-1-to-STR-2):
$\text{iter-cat}_{R \to R \to (R \to (R \to A)) \to (R \to A)} =$
$\mathrm{I}\, f : Z_{\{R\}} \to Z_{\{R\}} \to (Z_{\{R\}} \to (R \to A)) \to (R \to A)\ .$
$\forall\, m, n : Z_{\{R\}},\ g : Z_{\{R\}} \to (R \to A)\ .\ f\, m\, n\, g \simeq$
$(m > n \mapsto \epsilon \mid (f\, m\, (n-1)\, g)\ \text{cat}\ (g\, n))$

(iterated concatenation).

New target development: STR-3.

New development morphism: MON-over-COF-1-to-STR-3.

Notation for the application of

$$\text{set-cat}_{(\{R \to A\} \times \{R \to A\}) \to \{R \to A\}}$$

and

$$\text{iter-cat}_{R \to R \to (R \to (R \to A)) \to (R \to A)}$$

are defined in Table 13.

13 Related work

As we have seen, a theory (or development) graph provides an effective architecture for formalizing a body of mathematical knowledge. It is especially useful for creating a large library of formal mathematical knowledge that, by necessity, must be constructed in parallel by multiple developers. The library is built in parts by separate development teams and then the parts are linked together by morphisms. Mathematical knowledge is organized as a theory graph in several proof assistants and logical frameworks including Ergo [44], IMPS [22, 24], Isabelle [5], LF [53], MMT [52], and PVS [47]. Theory graphs are also employed in several software specification and development systems including ASL [57], CASL [3, 4], EHDM [55], Hets [40], IOTA [41], KIDS [58], OBJ [27], and Specware [59].

Simple type theory in the form of Church's type theory is a popular logic for formal mathematics. There are several proof assistants that implement versions of Church's type theory including HOL [29], HOL Light [31], IMPS [23, 24], Isabelle/HOL [48], ProofPower [51], PVS [46], and TPS [2]. As we mentioned in Section 1, the IMPS proof assistant is especially noteworthy here since it implements LUTINS [13, 14, 15], a version of Church's type theory that admits undefined expressions and is closely related to Alonzo.

In recent years, there has been growing interest in formalizing mathematics within dependent logics. Several proof assistants and programming languages are based on versions of dependent type theory including Agda [7, 45], Automath [43], Epigram [11], F* [12], Idris [34], Lean [10], Nuprl [9], and Rocq [54]. So which type theory is better for formal mathematics, simple type theory or dependent type theory? This question has become hotly contested. We hope that the reader will see our formalization of monoid theory in Alonzo as evidence for the efficacy of simple type theory as a logical basis for formal mathematics. The reader might also be interested in looking at these recent papers that advocate for simple type theory: [6, 49, 50].

Since monoid theory is a relatively simple subject, there have not been many attempts to formalize it by itself, but there have been several formalizations of group theory. Here are some examples: [26, 28, 35, 56, 60, 61].

There are two other important alternatives to the standard approach to formal mathematics. The first is Tom Hales' *formal abstracts in mathematics* project [25, 30] in which proof assistants are used to the create *formal abstracts*, which are formal presentations of mathematical theorems without formal proofs. The second is Michael Kohlhase's *flexiformal mathematics* [33, 36, 37] initiative in which mathematics is a mixture of traditional and formal mathematics and proofs can be either traditional or formal. The alternative approach we offer is similar to both of these approaches, but there are important differences. The formal abstracts approach seeks to formalize *collections of theorems* without proofs using proof assistants, while we seek to formalize *theory graphs* with either traditional or formal proofs using supporting software that can be much simpler than a proof assistant. The objective of the flexiformal mathematics approach is to give the user the flexibility to produce mathematics with varying degrees of formality. In contrast, our approach is to produce mathematics that is fully formal except for proofs.

14 Conclusion

The developments and development morphisms presented in Sections 4–12 form the development graph G_{mon} shown in Figure 1. The development graph shows all the development morphisms that we have explicitly defined (7 inclusions via theory extension modules and 10 noninclusions via theory and development definition modules) plus an implicit inclusion from COM-MON to COM-MON-over-COF. A development morphism that is an inclusion is designated by a $\hookrightarrow$ arrow and a noninclusion is designated by a $\rightarrow$ arrow. There are many, many more useful development morphisms that are not shown in G_{mon}, including implicit inclusions and a vast number of development morphisms into the theory COF.

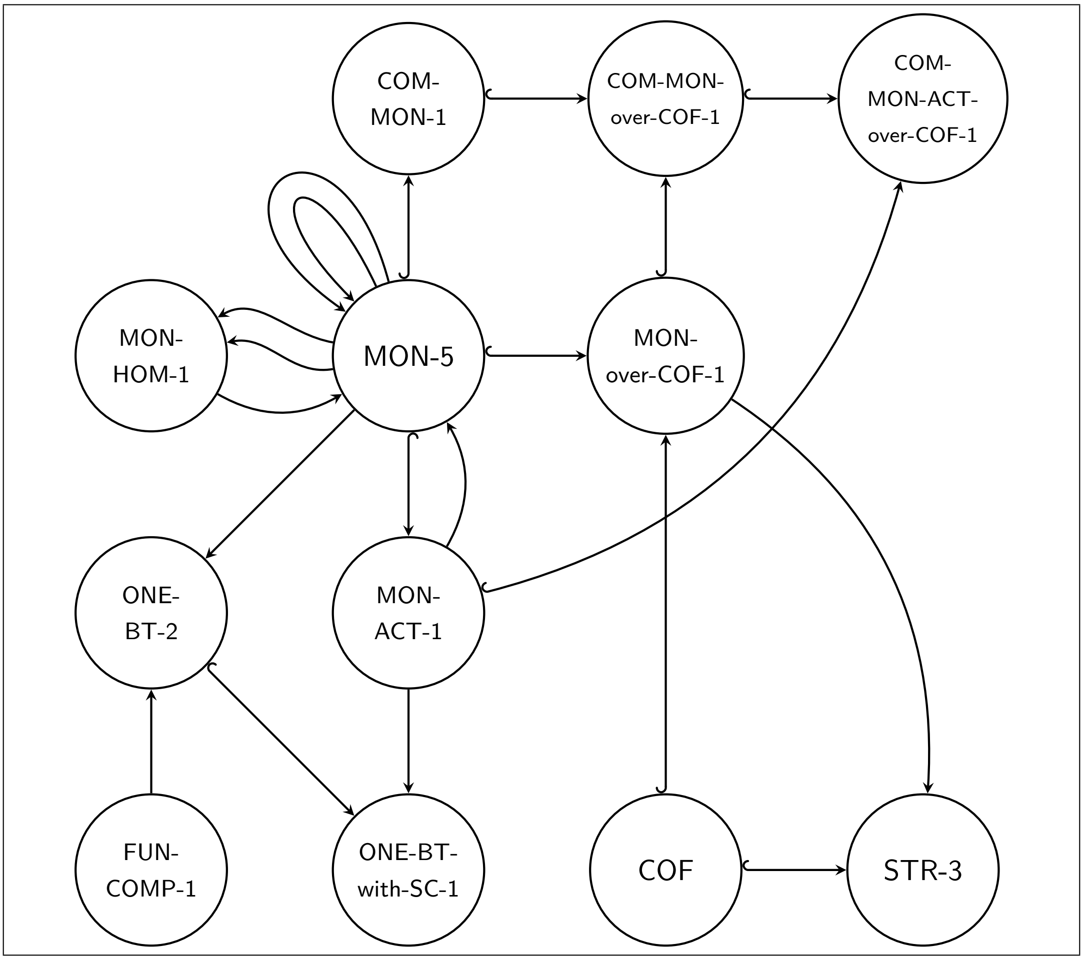


Figure 1: The Monoid Theory Development Graph

The construction of G_{mon} illustrates how a body of mathematical knowledge can be formalized in Alonzo as a development graph in accordance with the little theories method and the alternative approach. G_{mon} could be extended to include other mathematical concepts related to monoids such as categories. It could be incorporated in a development graph that formalizes a more extensive body of mathematical knowledge. And it could also be used as a foundation for building a formalization of group theory. This would be done by lifting each development D of a theory T that extends MON to a development D' of a theory T' that extends a theory GRP of groups obtained by adding an inverse operation to MON. The lifting of D to D' would include constructing inclusions from MON to GRP and from T to T' via theory extensions.

The formalization of monoid theory we have presented demonstrates three things. First, it demonstrates the power of the little theories method. The formalization is largely free of redundancy since each mathematical topic is articulated in just one development D, the development for the little theory that is optimal for the topic in level of abstraction and choice of vocabulary. If we create a translation Φ from D to another development D' and prove that Φ is a morphism, then we can freely transport the definitions and theorems of D to D' via Φ. That is, an abstract concept or fact that has been validated in D can be translated to a concrete instance of the concept or fact that is automatically validated in D' provided the translation is a morphism. (This is illustrated by our use of the development morphism MON-over-COF-to-STR-2 to transport definitions and theorems about monoids to a development about strings.) As the result, the same concept or fact can appear in many places in the theory graph but under different assumptions and involving different vocabulary. (For example, the notion of a submonoid represented by the constant $\mathsf{submonoid}_{\{M\}\rightarrow o}$ defined in MON-1 appears in ONE-BT-2 as the notion of a transformation monoid represented by the constant $\mathsf{trans\text{-}monoid}_{\{S\rightarrow S\}\rightarrow o}$.) In short, we have shown how the little theories method enables mathematical knowledge to be formalized to maximize clarity and minimize redundancy.

Second, the formalization demonstrates that the alternative approach to formal mathematics (with traditional and formal proofs) has two advantages over the standard approach (with only formal proofs): (1) communication is more effective since the user has greater freedom of expression and (2) formalization is easier since the approach offers greater accessibility. The standard approach is done with the help of a proof assistant and all proofs are formal and mechanically checked. Proof assistants are consequently very complex and notoriously difficult to learn how to use. Traditional proofs are easier to read and write than formal proofs and are better suited for communicating the ideas behind proofs. Moreover, since the alternative approach does not require a facility for developing and checking formal proofs, it can be done with software support that is much simpler and easier to use than a proof assistant. (In this paper, our software support was just a set of LaTeX macros and environments.)

Third, the formalization demonstrates that Alonzo is well suited for expressing and reasoning about mathematical ideas. The simple type theory machinery of Alonzo — function and product types, function application and abstraction, definite description, and ordered pairs — enables mathematical expressions to be formulated in a direct and natural manner. It also enables almost every single mathematical structure or set of similar mathematical structures to be specified by an Alonzo development. (For example, the development ONE-BT-2 specifies the set of mathematical structures consisting of a set S and the set $S \to S$ of transformations on S.) The admission of undefined expressions in Alonzo enables statements involving partial and total functions and definite descriptions to be expressed directly, naturally, and succinctly. (For example, if $M = (m, \cdot, e)$ is a monoid, the operation that makes a submonoid $m' \subseteq m$ of M a monoid itself is exactly what is expected: the partial function that results from restricting $\cdot$ to $m' \times m'$.) And the notational definitions and conventions employed in Alonzo enables mathematical expressions to be presented with largely the same notation that is used mathematical practice. (For example, Thm33: $\forall\, x : \mathsf{str} \,.\, \epsilon x = x \epsilon = x$, that states ϵ is an identity element for concatenation, is written just as one would expect it to be written in mathematical practice.)

We believe that this paper achieves our overarching goal: To demonstrate that mathematical knowledge can be very effectively formalized in a version of simple type theory like Alonzo using the little theories method and the alternative approach to formal mathematics. We also believe that it illustrates the benefits of employing the little theories method, the alternative approach, and Alonzo in formal mathematics.

A Validation of definitions and theorems

Let $D = (T, \Xi)$ be a development where T is the bottom theory of the development and $\Xi = [P_1, \ldots, P_n]$ is the list of definition and theorem packages of the development. For each i with $1 \leq i \leq n$, P_i has the form $(p, \mathbf{c}_\alpha, \mathbf{A}_\alpha, \pi)$ if P_i is a definition package and has the form $P_i = (p, \mathbf{A}_o, \pi)$ if P_i is a theorem package. Define $T_0 = T$ and, for all i with $0 \leq i \leq n-1$, define $T_{i+1} = T[P_{i+1}]$ if P_{i+1} is a definition package and $T_{i+1} = T_i$ if P_{i+1} is a theorem package. In the former case, π is a proof that $\mathbf{A}_\alpha\!\downarrow$ is valid in T_i, and in the latter case, π is a proof that $\mathbf{A}_o$ is valid in T_i. These proofs may be either traditional or formal. See Chapter 12 of [20] for further details.

The validation proofs for the definitions and theorems of a development are not included in the modules we have used to construct developments and to transport definitions and theorems. Instead, we give in this appendix, for each of the definitions and theorems in the developments defined in Sections 4–12, a traditional proof that validates the definition or theorem. The proofs are almost entirely straightforward. The proofs extensively reference the axioms, rules of inference, and metatheorems of $\mathfrak{A}$, the formal proof system for Alonzo presented in [20]. These are legitimate to use since $\mathfrak{A}$ is sound by the Soundness Theorem [20, Theorem B.11].

A.1 Development of MON

1. Thm1: $\mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$

   (models of MON define monoids).

   Proof of the theorem. Let $T = (L, \Gamma)$ be MON. We must show

   $(\star)\ T \vDash \mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$.

   $$\Gamma \vDash U_{\{M\}}\downarrow \quad (1)$$
   $$\Gamma \vDash U_{\{M\}} \neq \emptyset_{\{M\}} \quad (2)$$
   $$\Gamma \vDash \cdot_{(M \times M) \to M} \downarrow (U_{\{M\}} \times U_{\{M\}}) \to U_{\{M\}} \quad (3)$$
   $$\Gamma \vDash \mathsf{e}_M \downarrow U_{\{M\}} \quad (4)$$
   $$\Gamma \vDash \forall x, y, z : U_{\{M\}} \,.\, x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad (5)$$
   $$\Gamma \vDash \forall x : U_{\{M\}} \,.\, \mathsf{e} \cdot x = x \quad (6)$$
   $$\Gamma \vDash \mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M) \quad (7)$$

   (1) and (2) follow from parts 1 and 2, respectively, of Lemma B.1; (3) follows from [20, Axiom A5.2] and parts 8–10 of Lemma B.1; (4) follows from [20, Axiom A5.2] and part 8 of Lemma B.1; (5) and (6) follow from Axioms 1 and 2, respectively, of T and part 5 of Lemma B.1; and (7) follows from (1)–(6) and the definition of MONOID in Table 10. Therefore, $(\star)$ holds. □

2. Thm2: $\mathsf{TOTAL}(\cdot_{(M \times M) \to M})$ ($\cdot$ is total).

   Proof of the theorem. Let $\mathbf{A}_o$ be

   $\forall x : M \times M \,.\, (\cdot_{(M \times M) \to M}\, x)\downarrow$

   and $T = (L, \Gamma)$ be MON. TOTAL is the abbreviation introduced by the notational definition given in Table 7, and so $\mathsf{TOTAL}(\cdot_{(M \times M) \to M})$ stands for $\mathbf{A}_o$. Thus we must show $(\star)\ T \vDash \mathbf{A}_o$.

   $$\Gamma \vDash (x : M \times M)\downarrow \quad (1)$$
   $$\Gamma \vDash (x : M \times M) = (\mathsf{fst}\, x, \mathsf{snd}\, x) \quad (2)$$
   $$\Gamma \vDash (\mathsf{fst}\, x)\downarrow \wedge (\mathsf{snd}\, x)\downarrow \quad (3)$$
   $$\Gamma \vDash (\mathsf{fst}\, x) \cdot ((\mathsf{fst}\, x) \cdot (\mathsf{snd}\, x)) = ((\mathsf{fst}\, x) \cdot (\mathsf{fst}\, x)) \cdot (\mathsf{snd}\, x) \quad (4)$$
   $$\Gamma \vDash ((\mathsf{fst}\, x) \cdot (\mathsf{snd}\, x))\downarrow \quad (5)$$
   $$\Gamma \vDash (\cdot_{(M \times M) \to M}\, (\mathsf{fst}\, x, \mathsf{snd}\, x))\downarrow \quad (6)$$
   $$\Gamma \vDash \mathbf{A}_o \quad (7)$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from (1) and [20, Axiom A7.4] by Universal Instantiation [20, Theorem A.14]; (3) follows from (2) by [20, Axioms A5.5, A7.2, and A7.3]; (4) follows from (3) and Axiom 1 of T by Universal Instantiation [20, Theorem A.14]; (5) follows from (4) by [20, Axioms A5.4 and A5.10]; (6) follows from (5) by notational definition; and (7) follows from (6) by Universal Generalization [20, Theorem A.30] using (2) and the fact that $(x : (M \times M))$ is not free in Γ since Γ is a set of sentences. □

3. Thm3: $\forall\, x : M \,.\, (\forall\, y : M \,.\, x \cdot y = y \cdot x = y) \Rightarrow x = \mathsf{e}$ (uniqueness of identity element).

Proof of the theorem. Let $\mathbf{A}_o$ be

$$\forall\, y : M \,.\, (x : M) \cdot y = y \cdot (x : M) = y$$

and $T = (L, \Gamma)$ be MON. We must show $(\star)$ $T \vDash \forall\, x : M \,.\, \mathbf{A}_o \Rightarrow x = \mathsf{e}$.

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \mathsf{e}{\downarrow} \tag{1}$$
$$\Gamma \cup \{\mathbf{A}_o\} \vDash (x : M){\downarrow} \tag{2}$$
$$\Gamma \cup \{\mathbf{A}_o\} \vDash (x : M) \cdot \mathsf{e} = \mathsf{e} \cdot (x : M) = \mathsf{e} \tag{3}$$
$$\Gamma \cup \{\mathbf{A}_o\} \vDash \mathsf{e} \cdot (x : M) = (x : M) \cdot \mathsf{e} = (x : M) \tag{4}$$
$$\Gamma \cup \{\mathbf{A}_o\} \vDash (x : M) = \mathsf{e} \tag{5}$$
$$\Gamma \vDash \mathbf{A}_o \Rightarrow (x : M) = \mathsf{e} \tag{6}$$
$$\Gamma \vDash \forall\, x : M \,.\, \mathbf{A}_o \Rightarrow x = \mathsf{e} \tag{7}$$

(1) follows from constants always being defined by [20, Axiom A5.2]; (2) follows from variables always being defined by [20, Axiom A5.1]; (3) follows (1) and $\mathbf{A}_o$ by Universal Instantiation [20, Theorem A.14]; (4) follows (2) and Axiom 2 of T by Universal Instantiation; (5) follows from (3) and (4) by the Equality Rules [20, Lemma A.13]; (6) follows from (5) by the Deduction Theorem [20, Lemma A.50]; and (7) follows from (6) by Universal Generalization [20, Theorem A.30] using the fact that $(x : M)$ is not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

4. Def1: $\mathsf{submonoid}_{\{M\} \to o} =$
$\lambda\, s : \{M\} \,.\, s \neq \emptyset_{\{M\}} \wedge (\cdot|_{s \times s} \downarrow (s \times s) \to s) \wedge \mathsf{e} \in s$ (submonoid).

Proof that RHS is defined. Let $\mathbf{A}_{\{M\} \to o}$ be the RHS of Def1. We must show that MON $\vDash \mathbf{A}_{\{M\} \to o}{\downarrow}$. This follows immediately from function abstractions always being defined by [20, Axiom A5.11]. □

5. Thm4: $\forall\, s : \{M\}\ .\ \mathsf{submonoid}\, s \Rightarrow \mathsf{MONOID}(s, \cdot|_{s\times s}, \mathsf{e})$ (submonoids are monoids).

Proof of the theorem. Let $\mathbf{A}_o$ be

$$\mathsf{submonoid}(s)$$

and $T = (L, \Gamma)$ be MON extended by Def1. We must show

$$(\star)\ T \vDash \forall\, s : \{M\}\ .\ \mathbf{A}_o \Rightarrow \mathsf{MONOID}(s, \cdot|_{(s\times s)}, \mathsf{e}).$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash s_{\{M\}}\downarrow \quad (1)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash s \neq \emptyset_{\{M\}} \quad (2)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \cdot|_{(s\times s)} \downarrow (s \times s) \rightarrow s \quad (3)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \mathsf{e} \in s \quad (4)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \mathsf{e} \downarrow s \quad (5)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \forall\, x, y, z : s\ .\ \cdot|_{(s\times s)}(x, \cdot|_{(s\times s)}(y, z)) = \cdot|_{(s\times s)}(\cdot|_{(s\times s)}(x, y), z) \quad (6)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \forall\, x : s\ .\ \cdot|_{(s\times s)}(\mathsf{e}, x) = \cdot|_{(s\times s)}(x, \mathsf{e}) = x \quad (7)$$

$$\Gamma \cup \{\mathbf{A}_o\} \vDash \mathsf{MONOID}(s, \cdot|_{(s\times s)}, \mathsf{e}) \quad (8)$$

$$\Gamma \vDash \mathbf{A}_o \Rightarrow \mathsf{MONOID}(s, \cdot|_{(s\times s)}, \mathsf{e}) \quad (9)$$

$$\Gamma \vDash \forall\, s : \{M\}\ .\ \mathbf{A}_o \Rightarrow \mathsf{MONOID}(s, \cdot|_{(s\times s)}, \mathsf{e}) \quad (10)$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2), (3), and (4) follow directly from Def1; (5) follows from [20, Axiom A5.2] and (4); (6) and (7) follow from Thm1, $\cdot|_{(s\times s)} \sqsubseteq \cdot_{(M\times M)\rightarrow M}$, and the fact that $\cdot|_{(s\times s)}$ is total on $s \times s$ by Thm2; (8) follows from (1)–(3) and (5)–(7) by the definition of MONOID in Table 10; (9) follows from (8) by the Deduction Theorem [20, Theorem A.50]; and (10) follows from (9) by Universal Generalization [20, Theorem A.30] using the fact that $(s : \{M\})$ is not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

6. Thm5: $\mathsf{submonoid}\, \{\mathsf{e}\}$ (minimum submonoid).

Proof of the theorem. Let $T = (L, \Gamma)$ be MON extended by Def1. We must show $(\star)\ T \vDash \mathsf{submonoid}\,\{\mathsf{e}\}$.

$$\Gamma \vDash \mathsf{e} \in \{\mathsf{e}\} \tag{1}$$
$$\Gamma \vDash \{\mathsf{e}\} \neq \emptyset_{\{M\}} \tag{2}$$
$$\Gamma \vDash \mathsf{e} \cdot \mathsf{e} = \mathsf{e} \tag{3}$$
$$\Gamma \vDash \cdot|_{\{\mathsf{e}\}\times\{\mathsf{e}\}} \downarrow (\{\mathsf{e}\} \times \{\mathsf{e}\}) \to \{\mathsf{e}\} \tag{4}$$
$$\Gamma \vDash \mathsf{submonoid}\,\{\mathsf{e}\} \tag{5}$$

(1) is trivial; (2) follows from (1) because $\{\mathsf{e}\}$ has at least one member; (3) follows from Axiom 2 of T by Universal Instantiation [20, Theorem A.14]; (4) follows directly from (1), (3), and the fact that the only member of $\{\mathsf{e}\}$ is e; and (5) follows from (1), (2), (4), and Def1. Therefore, $(\star)$ holds. □

7. Thm6: $\mathsf{submonoid}\,U_{\{M\}}$ (maximum submonoid).

   Proof of the theorem. Let $T = (L, \Gamma)$ be MON extended by Def1. We must show $(\star)\ T \vDash \mathsf{submonoid}\,U_{\{M\}}$.

$$\Gamma \vDash \mathsf{MONOID}(U_{\{M\}}, \cdot_{(M\times M)\to M}, \mathsf{e}) \tag{1}$$
$$\Gamma \vDash U_{\{M\}} \neq \emptyset_{\{M\}} \wedge \mathsf{e} \in U_{\{M\}} \tag{2}$$
$$\Gamma \vDash \cdot|_{U_{\{M\}}\times U_{\{M\}}} \downarrow (U_{\{M\}} \times U_{\{M\}}) \to U_{\{M\}} \tag{3}$$
$$\Gamma \vDash \mathsf{submonoid}\,U_{\{M\}} \tag{4}$$

   (1) is Thm1; (2) follows immediately from (1); (3) follows from (1) by part 12 of Lemma B.1; and (4) follows from (1), (2), (3), and Def1. Therefore, $(\star)$ holds. □

8. Def2: $\cdot^{\mathrm{op}}_{(M\times M)\to M} = \lambda\, p : M \times M \,.\, (\mathsf{snd}\, p) \cdot (\mathsf{fst}\, p)$ (opposite of $\cdot$).

   Proof that RHS is defined. Similar to the proof that the RHS of Def1 is defined. □

9. Thm7: $\forall\, x, y, z : M \,.\, x \cdot^{\mathrm{op}} (y \cdot^{\mathrm{op}} z) = (x \cdot^{\mathrm{op}} y) \cdot^{\mathrm{op}} z$ ($\cdot^{\mathrm{op}}$ is associative).

   Proof of the theorem. Let $\mathbf{A}_o$ be

$$x \cdot^{\mathrm{op}} (y \cdot^{\mathrm{op}} z) = (x \cdot^{\mathrm{op}} y) \cdot^{\mathrm{op}} z$$

   and $T = (L, \Gamma)$ be MON extended by Def2. We must show

$$(\star)\ T \vDash \forall\, x, y, z : M \,.\, \mathbf{A}_o.$$

$$\Gamma \models (x : M)\downarrow \wedge (y : M)\downarrow \wedge (z : M)\downarrow \quad (1)$$
$$\Gamma \models (z \cdot y) \cdot x = z \cdot (y \cdot x) \quad (2)$$
$$\Gamma \models \mathbf{A}_o \quad (3)$$
$$\Gamma \models \forall\, x, y, z : M \mathrel{.} \mathbf{A}_o \quad (4)$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from (1) and Axiom 1 of T by Universal Instantiation [20, Theorem A.14] and the Equality Rules [20, Theorem A.13]; (3) follows from Lemma B.2 and (2) by repeated applications of Rule R2′ [20, Lemma A.2] using (⋆⋆) the fact that $(x : M)$, $(y : M)$, and $(z : M)$ are not free in Γ since Γ is a set of sentences; and (4) follows from (3) by Universal Generalization [20, Theorem A.30] again using (⋆⋆). Therefore (⋆) holds. □

10. Thm8: $\forall\, x : M \mathrel{.} \mathsf{e} \cdot^{\mathrm{op}} x = x \cdot^{\mathrm{op}} \mathsf{e} = x$
(e is an identity element with respect to $\cdot^{\mathrm{op}}$).

Proof of the theorem. Let $\mathbf{A}_o$ be

$$\mathsf{e} \cdot^{\mathrm{op}} x = x \cdot^{\mathrm{op}} \mathsf{e} = x$$

and $T = (L, \Gamma)$ be MON extended by Def2. We must show

(⋆) $T \models \forall\, x : M \mathrel{.} \mathbf{A}_o$.

$$\Gamma \models (x : M)\downarrow \quad (1)$$
$$\Gamma \models x \cdot \mathsf{e} = \mathsf{e} \cdot x = x \quad (2)$$
$$\Gamma \models \mathbf{A}_o \quad (3)$$
$$\Gamma \models \forall\, x : M \mathrel{.} \mathbf{A}_o \quad (4)$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from (1) and Axiom 2 of T by Universal Instantiation [20, Theorem A.14] and the Equality Rules [20, Theorem A.13]; (3) follows from Lemma B.2 and (2) by repeated applications of Rule R2′ [20, Lemma A.2] using (⋆⋆) the fact that $(x : M)$ is not free in Γ since Γ is a set of sentences; and (4) follows from (3) by Universal Generalization [20, Theorem A.30] again using (⋆⋆). Therefore (⋆) holds. □

11. Def3: $\odot_{(\{M\}\times\{M\})\to\{M\}} = \mathsf{set\text{-}op}_{((M\times M)\to M)\to((\{M\}\times\{M\})\to\{M\})}$.
(set product).

Proof that RHS is defined. Let $\mathbf{A}_{(\{M\} \times \{M\}) \rightarrow \{M\}}$ be the RHS of Def3. We must show ($\star$) MON $\models \mathbf{A}_{(\{M\} \times \{M\}) \rightarrow \{M\}}\downarrow$. Since constants are always defined by [20, Axiom A5.2], $\mathbf{A}_{(\{M\} \times \{M\}) \rightarrow \{M\}}$ beta-reduces to a function abstraction by [20, Axiom A4]. Since every function abstraction is defined by [20, Axiom A5.11], we have ($\star$) by Quasi-Equality Substitution [20, Lemma A.2]. □

12. Def4: $\mathsf{E}_{\{M\}} = \{\mathsf{e}_M\}$ (set identity element).

Proof that RHS is defined. We must show ($\star$) MON $\models \{\mathsf{e}_M\}\downarrow$. Now $\{\mathsf{e}_M\}$ stands for

$$(\lambda\, x_1 : M \,.\, \lambda\, x : M \,.\, x = x_1)\,(\mathsf{e}_M).$$

Since constants are always defined by [20, Axiom A5.2], $\{\mathsf{e}_M\}$ beta-reduces to

$$\lambda\, x : M \,.\, x = \mathsf{e}_M$$

by [20, Axiom A4]. Since every function abstraction is defined by [20, Axiom A5.11], we have ($\star$) by Quasi-Equality Substitution [20, Lemma A.2]. □

13. Thm9: $\forall\, x, y, z : \{M\} \,.\, x \odot (y \odot z) = (x \odot y) \odot z$ ($\odot$ is associative).

Proof of the theorem. Let $T = (L, \Gamma)$ be MON extended by Def3. We must show

$$(\star)\ T \models \forall\, x, y, z : \{M\} \,.\, x \odot (y \odot z) = (x \odot y) \odot z.$$

$$\Gamma \models (x : \{M\})\downarrow \wedge (y : \{M\})\downarrow \wedge (z : \{M\})\downarrow \quad (1)$$

$$\Gamma \models x \odot (y \odot z) = \{d : M \mid \exists\, a : x,\, b : y,\, c : z \,.\, d = a \cdot (b \cdot c)\} \quad (2)$$

$$\Gamma \models (x \odot y) \odot z = \{d : M \mid \exists\, a : x,\, b : y,\, c : z \,.\, d = (a \cdot b) \cdot c\} \quad (3)$$

$$\Gamma \models x \odot (y \odot z) = (x \odot y) \odot z \quad (4)$$

$$\Gamma \models \forall\, x, y, z : \{M\} \,.\, x \odot (y \odot z) = (x \odot y) \odot z \quad (5)$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (3) follow from (1) and Def3; (4) follows from (2) and (3) by Axiom 1 of T; and (5) follows from (4) by Universal Generalization [20, Theorem A.30] using the fact that x, y, and z are not free in Γ since Γ is a set of sentences. Therefore, ($\star$) holds. □

14. $\mathsf{Thm10}$: $\forall\, x : \{M\} \,.\, \mathsf{E} \odot x = x \odot \mathsf{E} = x$

(E is an identity element with respect to $\odot$).

Proof of the theorem. Let $T = (L, \Gamma)$ be MON extended by Def3 and Def4. We must show

$$(\star)\ T \vDash \forall\, x : \{M\} \,.\, \mathsf{E} \odot x = x \odot \mathsf{E} = x.$$

$$\Gamma \vDash (x : \{M\})\!\downarrow \tag{1}$$
$$\Gamma \vDash \mathsf{E}\!\downarrow \tag{2}$$
$$\Gamma \vDash \mathsf{E} \odot x = \{b : M \mid \exists\, a : x \,.\, b = \mathsf{e} \cdot a\} \tag{3}$$
$$\Gamma \vDash x \odot \mathsf{E} = \{b : M \mid \exists\, a : x \,.\, b = a \cdot \mathsf{e}\} \tag{4}$$
$$\Gamma \vDash \mathsf{E} \odot x = x \odot \mathsf{E} = x \tag{5}$$
$$\Gamma \vDash \forall\, x : \{M\} \,.\, \mathsf{E} \odot x = x \odot \mathsf{E} = x \tag{6}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from constants always defined by [20, Axiom A5.2]; (3) and (4) follow from (1), (2), Def3, and Def4; (5) follows from (3) and (4) by Axiom 2 of T; and (6) follows from (5) by Universal Generalization [20, Theorem A.30] using the fact that x is not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

15. Thm11 (Thm1-via-MON-to-opposite-monoid):
$\mathsf{MONOID}(U_{\{M\}}, \cdot^{\mathrm{op}}_{(M \times M) \to M}, \mathsf{e}_M)$ (opposite monoids are monoids).

Proof of the theorem. Let T be the top theory of MON-1. We must show $T \vDash$ Thm11. We have previously proved $(\star)$ MON $\vDash$ Thm1. $\Phi =$ MON-to-opposite-monoid is a development morphism from MON to MON-1, and so $\widetilde{\Phi} = (\mu, \nu)$ is a theory morphism from MON to T. Thus $(\star)$ implies $T \vDash \nu(\mathsf{Thm1})$. Therefore, $T \vDash$ Thm11 since $\mathsf{Thm11} = \nu(\mathsf{Thm1})$. □

16. Thm12 (Thm1-via-MON-to-set-monoid):
$\mathsf{MONOID}(U_{\{\{M\}\}}, \odot_{(\{M\} \times \{M\}) \to \{M\}}, \mathsf{E}_{\{M\}})$

(set monoids are monoids).

Proof of the theorem. Similar to the proof of Thm11. □

A.2 Development of COM-MON

1. Thm13: $\mathsf{COM\text{-}MONOID}(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M)$

(models of COM-MON define commutative monoids).

Proof of the theorem. Let $T = (L, \Gamma)$ be $\mathsf{COM\text{-}MON}$. We must show

$(\star)$ $T \models \mathsf{COM\text{-}MONOID}(U_{\{M\}}, \cdot_{(M \times M) \rightarrow M}, \mathsf{e}_M)$.

$$\Gamma \models \mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \rightarrow M}, \mathsf{e}_M) \tag{1}$$
$$\Gamma \models \forall\, x, y : U_{\{M\}} \,.\, x \cdot y = y \cdot x \tag{2}$$

(1) follows from $\mathsf{MON} \leq T$ and the fact that $\mathsf{Thm1}$ is a theorem of MON; and (2) follows from Axiom 3 of T and part 5 of Lemma B.1. Therefore, $(\star)$ follows from (1), (2), and the notational definition of $\mathsf{COM\text{-}MONOID}$ given in Table 10. □

2. $\mathsf{Def5}$: $\leq_{M \rightarrow M \rightarrow o} = \lambda\, x, y : M \,.\, \exists\, z : M \,.\, x \cdot z = y$ (weak order).

   Proof that RHS is defined. Similar to the proof that the RHS of $\mathsf{Def1}$ is defined. □

3. $\mathsf{Thm14}$: $\forall\, x : M \,.\, x \leq x$ (reflexivity).

   Proof of the theorem. Let $T = (L, \Gamma)$ be $\mathsf{COM\text{-}MON}$ extended by $\mathsf{Def5}$. We must show

   $(\star)$ $T \models \forall\, x : M \,.\, x \leq x$.

$$\Gamma \models (x : M){\downarrow} \tag{1}$$
$$\Gamma \models (x \leq x) \simeq (\exists\, z : M \,.\, x \cdot z = x) \tag{2}$$
$$\Gamma \models x \cdot \mathsf{e} = x \tag{3}$$
$$\Gamma \models \exists\, z : M \,.\, x \cdot z = x \tag{4}$$
$$\Gamma \models x \leq x \tag{5}$$
$$\Gamma \models \forall\, x : M \,.\, x \leq x \tag{6}$$

   (1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from $\mathsf{Def5}$ and Extensionality [20, Axiom A3] using the Substitution Rule [20, Theorem A.31] and Beta-Reduction [20, Axiom A4]; (3) follows from (1) and Axiom 2 of T by Universal Instantiation [20, Theorem A.14]; (4) follows from (3) by Existential Generalization [20, Theorem A.51]; (5) follows from (2) and (4) by Rule R2$'$ [20, Lemma A.2]; and (6) follows from (5) by Universal Generalization [20, Theorem A.30] using the fact that x is not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

4. Thm15: $\forall\, x, y, z : M \,.\, (x \leq y \wedge y \leq z) \Rightarrow x \leq z$ (transitivity).

Proof of the theorem. Let $\mathbf{A}_o$ be $(x \leq y \wedge y \leq z)$, $\mathbf{B}_o$ be $x \cdot u = y$, and $\mathbf{C}_o$ be $y \cdot v = z$ (where these variables all have type M). Also let $T = (L, \Gamma)$ be COM-MON extended by Def5. We must show

$(\star)$ $T \vDash \forall\, x, y, z : M \,.\, \mathbf{A}_o \Rightarrow x \leq z.$

$$\Gamma \cup \{\mathbf{B}_o, \mathbf{C}_o\} \vDash (x : M)\!\downarrow \wedge\, (y : M)\!\downarrow \wedge\, (z : M)\!\downarrow \wedge\, (u : M)\!\downarrow \wedge\, (v : M)\!\downarrow \tag{1}$$

$$\Gamma \cup \{\mathbf{B}_o, \mathbf{C}_o\} \vDash (x \cdot u) \cdot v = z \tag{2}$$

$$\Gamma \cup \{\mathbf{B}_o, \mathbf{C}_o\} \vDash (x \cdot u) \cdot v = x \cdot (u \cdot v) \tag{3}$$

$$\Gamma \cup \{\mathbf{B}_o, \mathbf{C}_o\} \vDash x \cdot (u \cdot v) = z \tag{4}$$

$$\Gamma \cup \{\mathbf{B}_o, \mathbf{C}_o\} \vDash \exists\, w : M \,.\, x \cdot w = z \tag{5}$$

$$\Gamma \cup \{\mathbf{B}_o\} \vDash (y \cdot v = z) \Rightarrow (\exists\, w : M \,.\, x \cdot w = z) \tag{6}$$

$$\Gamma \cup \{\mathbf{B}_o\} \vDash (\exists\, v : M \,.\, y \cdot v = z) \Rightarrow (\exists\, w : M \,.\, x \cdot w = z) \tag{7}$$

$$\Gamma \vDash (x \cdot u = y) \Rightarrow ((\exists\, v : M \,.\, y \cdot v = z) \Rightarrow (\exists\, w : M \,.\, x \cdot w = z)) \tag{8}$$

$$\Gamma \vDash (\exists\, u : M \,.\, x \cdot u = y) \Rightarrow ((\exists\, v : M \,.\, y \cdot v = z) \Rightarrow (\exists\, w : M \,.\, x \cdot w = z)) \tag{9}$$

$$\Gamma \vDash x \leq y \Rightarrow (y \leq z \Rightarrow x \leq z) \tag{10}$$

$$\Gamma \vDash \mathbf{A}_o \Rightarrow x \leq z \tag{11}$$

$$\Gamma \vDash \forall\, x, y, z : M \,.\, \mathbf{A}_o \Rightarrow x \leq z \tag{12}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from $\mathbf{B}_o$ and $\mathbf{C}_o$ by the Equality Rules [20, Lemma A.13]; (3) follows from Axiom 1 of T by Universal Instantiation [20, Theorem A.14]; (4) follows from (2) and (3) by the Equality Rules [20, Lemma A.13]; (5) follows from (1), (4), and Thm2 by Existential Generalization [20, Theorem A.51]; (6) and (8) follow from (5) and (7), respectively, by the Deduction Theorem [20, Theorem A.50]; (7) and (9) follow from (6) and (8), respectively, by Existential Instantiation [20, Theorem A.52]; (10) follows from (1), (9), and Def5 by Beta-Reduction [20, Axiom A4] and Alpha-Conversion [20, Theorem A.18]; (11) follows from (10) by the Tautology Rule [20, Corollary A.46]; and (12) follows from (11) by Universal Generalization [20, Theorem A.30] using the fact that x, y, and z are not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

A.3 Development of FUN-COMP

1. Thm16: $\forall f : A \to B,\, g : B \to C,\, h : C \to D \,.\, f \circ (g \circ h) = (f \circ g) \circ h$
   ($\circ$ is associative).

   Proof of the theorem. Let $\mathbf{A}_o$ be the theorem and $T = (L, \Gamma)$ be FUN-COMP. We must show $(\star)$ $T \vDash \mathbf{A}_o$.

$$\Gamma \vDash (f : A \to B)\!\downarrow \wedge\, (g : B \to C)\!\downarrow \wedge\, (h : C \to D)\!\downarrow \wedge\, (x : A)\!\downarrow \quad (1)$$
$$\Gamma \vDash ((f \circ g) \circ h)\, x \simeq h\,(g\,(f\,x)) \quad (2)$$
$$\Gamma \vDash (f \circ (g \circ h))\, x \simeq h\,(g\,(f\,x)) \quad (3)$$
$$\Gamma \vDash ((f \circ g) \circ h)\, x \simeq (f \circ (g \circ h))\, x \quad (4)$$
$$\Gamma \vDash \forall\, x : A \,.\, (f \circ (g \circ h))\, x \simeq ((f \circ g) \circ h)\, x \quad (5)$$
$$\Gamma \vDash f \circ (g \circ h) = (f \circ g) \circ h \quad (6)$$
$$\Gamma \vDash \mathbf{A}_o \quad (7)$$

   (1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (3) both follow from (1), the definition of $\circ$ in Table 9, function abstractions are always defined by [20, Axiom A5.11], ordered pairs of defined components are always defined by [20, Axiom A7.1], Beta-Reduction [20, Axiom A4], and Quasi-Equality Substitution [20, Lemma A.2]; (4) follows from (2) and (3) by the Quasi-Equality Rules [20, Lemma A.4]; (5) follows from (4) by Universal Generalization [20, Theorem A.30] using the fact that x is not free in Γ since Γ is a set of sentences; (6) follows from (5) by Extensionality [20, Axiom A3]; and (7) follows from (6) by Universal Generalization using the fact that f, g, and h are not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

2. Thm17: $\forall f : A \to B \,.\, \mathsf{id}_{A \to A} \circ f = f \circ \mathsf{id}_{B \to B} = f$
   (identity functions are left and right identity elements).

   Proof of the theorem. Let $\mathbf{A}_o$ be the theorem and $T = (L, \Gamma)$ be

FUN-COMP. We must show ($\star$) $T \vDash \mathbf{A}_o$.

$$\Gamma \vDash (f : A \to B)\downarrow \wedge (x : A)\downarrow \tag{1}$$
$$\Gamma \vDash (\mathsf{id}_{A \to A} \circ f)\, x \simeq f\, x \tag{2}$$
$$\Gamma \vDash (f \circ \mathsf{id}_{B \to B})\, x \simeq f\, x \tag{3}$$
$$\Gamma \vDash \forall\, x : A \,.\, (\mathsf{id}_{A \to A} \circ f)\, x \simeq f\, x \tag{4}$$
$$\Gamma \vDash \forall\, x : A \,.\, (f \circ \mathsf{id}_{B \to B})\, x \simeq f\, x \tag{5}$$
$$\Gamma \vDash \mathsf{id}_{A \to A} \circ f = f \tag{6}$$
$$\Gamma \vDash f \circ \mathsf{id}_{B \to B} = f \tag{7}$$
$$\Gamma \vDash \mathsf{id}_{A \to A} \circ f = f \circ \mathsf{id}_{B \to B} = f \tag{8}$$
$$\Gamma \vDash \mathbf{A}_o \tag{9}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (3) both follow from (1), the definitions of id and $\circ$ in Table 9, function abstractions are always defined by [20, Axiom A5.11], ordered pairs of defined components are always defined by [20, Axiom A7.1], Beta-Reduction [20, Axiom A4], and Quasi-Equality Substitution [20, Lemma A.2]; (4) and (5) both follow from (2) and (3), respectively, by Universal Generalization [20, Theorem A.30] using the fact that x is not free in Γ since Γ is a set of sentences; (6) and (7) follow from (4) and (5), respectively, by Extensionality [20, Axiom A3]; (8) follows from (6) and (7) by the Equality Rules [20, Lemma A.13]; and (9) follows from (8) by Universal Generalization using the fact that f is not free in Γ since Γ is a set of sentences. Therefore, ($\star$) holds. □

A.4 Development of ONE-BT

1. Thm18 (Thm16-via-FUN-COMP-to-ONE-BT):
   $\forall\, f, g, h : S \to S \,.\, f \circ (g \circ h) = (f \circ g) \circ h$ ($\circ$ is associative).

   Proof of the theorem. Similar to the proof of Thm11. □

2. Thm19 (Thm17-via-FUN-COMP-to-ONE-BT):
   $\forall\, f : S \to S \,.\, \mathsf{id}_{S \to S} \circ f = f \circ \mathsf{id}_{S \to S} = f$
   ($\mathsf{id}_{S \to S}$ is an identity element with respect to $\circ$).

   Proof of the theorem. Similar to the proof of Thm11. □

3. Def6 (Def1-via-MON-to-ONE-BT):
   $\mathsf{trans\text{-}monoid}_{\{S \to S\} \to o} =$
   $\lambda\, s : \{S \to S\} \,.$

$s \neq \emptyset_{\{S \to S\}} \wedge$
$\mathsf{TOTAL\text{-}ON}(\circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{s \times s}, s \times s, s) \wedge$
$\mathsf{id}_{S \to S} \in s$ (transformation monoid).

Proof that RHS is defined. Let $\mathbf{A}^1_{\{M\} \to o}$ be the RHS of Def1, $\mathbf{A}^2_{\{S \to S\} \to o}$ be the RHS of Def6, T_1 be MON, and T_2 be ONE-BT, the top theory of ONE-BT-1. We must show $T_2 \models \mathbf{A}^2_{\{S \to S\} \to o}\downarrow$. We have previously proved $(\star)$ $T_1 \models \mathbf{A}^1_{\{M\} \to o}\downarrow$. MON-to-ONE-BT $= (\mu, \nu)$ is a theory morphism from T_1 to T_2. Thus $(\star)$ implies $T_2 \models \nu(\mathbf{A}^1_{\{M\} \to o}\downarrow)$. Therefore, $T_2 \models \mathbf{A}^2_{\{S \to S\} \to o}\downarrow$ since $\mathbf{A}^2_{\{S \to S\} \to o} = \nu(\mathbf{A}^1_{\{M\} \to o})$. □

4. Thm20 (Thm4-via-MON-to-ONE-BT):
   $\forall\, s : \{S \to S\}\ .$
   $\mathsf{trans\text{-}monoid}\ s \Rightarrow \mathsf{MONOID}(s, \circ_{((S \to S) \times (S \to S)) \to (S \to S)}|_{s \times s}, \mathsf{id}_{S \to S})$
   (transformation monoids are monoids).

   Proof of the theorem. Similar to the proof of Thm11. □

A.5 Development of MON-ACT

1. Thm21: $\mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{S\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S})$
   (models of MON-ACT define monoid actions).

   Proof of the theorem. Let $T = (L, \Gamma)$ be MON-ACT. We must show

   $(\star)\ T \models \mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{S\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S}).$

$$\Gamma \models \mathsf{MONOID}(U_{\{M\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M) \quad (1)$$
$$\Gamma \models U_{\{S\}}\downarrow \quad (2)$$
$$\Gamma \models U_{\{S\}} \neq \emptyset_{\{S\}} \quad (3)$$
$$\Gamma \models \mathsf{act}_{(M \times S) \to S} \downarrow (U_{\{M\}} \times U_{\{S\}}) \to U_{\{S\}} \quad (4)$$
$$\Gamma \models \forall\, x, y : U_{\{M\}},\ s : U_{\{S\}}\ .\ x \ \mathsf{act}\ (y \ \mathsf{act}\ s) = (x \cdot y)\ \mathsf{act}\ s \quad (5)$$
$$\Gamma \models \forall\, s : U_{\{S\}}\ .\ \mathsf{e}\ \mathsf{act}\ s = s \quad (6)$$
$$\Gamma \models \mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{S\}}, \cdot_{(M \times M) \to M}, \mathsf{e}_M, \mathsf{act}_{(M \times S) \to S}) \quad (7)$$

   (1) follows from MON $\models$ Thm1 and MON $\leq T$; (2) and (3) follow from parts 1 and 2, respectively, of Lemma B.1; (4) follows from [20, Axiom 5.2] and parts 8–10 of Lemma B.1; (5) and (6) follow from Axioms 3 and 4, respectively, of T and part 5 of Lemma B.1; and (7) follows from (1)–(6) and the definition of MON-ACTION in Table 10. Therefore, $(\star)$ holds. □

Thm22: $\mathsf{TOTAL}(\mathsf{act}_{(M \times S) \to S})$ (act is total).

Proof of the theorem. Let $T = (L, \Gamma)$ be MON-ACT.

$$T \vDash \mathsf{TOTAL}(\mathsf{act}_{(M \times S) \to S})$$

follows from Axiom 3 of T in the same way that $T \vDash \mathsf{TOTAL}(\cdot_{(M \times M) \to M})$ follows from Axiom 1 of MON as shown in the proof of Thm2. □

2. Def7: $\mathsf{orbit}_{S \to \{S\}} = \lambda\, s : S \,.\, \{t : S \mid \exists\, x : M \,.\, x \;\mathsf{act}\; s = t\}$ (orbit).

   Proof that RHS is defined. Similar to the proof that the RHS of Def1 is defined. □

3. Def8: $\mathsf{stabilizer}_{S \to \{M\}} = \lambda\, s : S \,.\, \{x : M \mid x \;\mathsf{act}\; s = s\}$ (stabilizer).

   Proof that RHS is defined. Similar to the proof that the RHS of Def1 is defined. □

4. Thm23: $\forall\, s : S \,.\, \mathsf{submonoid}\,(\mathsf{stabilizer}\, s)$ (stabilizers are submonoids).

   Proof of the theorem. Let $T = (L, \Gamma)$ be MON-ACT extended by Def7 and Def8. We must show

   $(\star)\ T \vDash \forall\, s : S \,.\, \mathsf{submonoid}\,(\mathsf{stabilizer}\, s)$.

$$\Gamma \vDash (s : S)\downarrow \tag{1}$$
$$\Gamma \vDash \mathsf{e}_M\downarrow \tag{2}$$
$$\Gamma \vDash (\mathsf{stabilizer}\, s) = \{x : M \mid x \;\mathsf{act}\; s = s\} \tag{3}$$
$$\Gamma \vDash \mathsf{e} \in (\mathsf{stabilizer}\, s) \tag{4}$$
$$\Gamma \vDash (\mathsf{stabilizer}\, s) \neq \emptyset_{\{M\}} \tag{5}$$
$$\Gamma \vDash \cdot|_{(\mathsf{stabilizer}\, s) \times (\mathsf{stabilizer}\, s)} \downarrow ((\mathsf{stabilizer}\, s) \times (\mathsf{stabilizer}\, s)) \to (\mathsf{stabilizer}\, s) \tag{6}$$
$$\Gamma \vDash (\mathsf{stabilizer}\, s)\downarrow \tag{7}$$
$$\Gamma \vDash \mathsf{submonoid}\,(\mathsf{stabilizer}\, s) = (\mathsf{stabilizer}\, s) \neq \emptyset_{\{M\}} \wedge \cdot|_{(\mathsf{stabilizer}\, s) \times (\mathsf{stabilizer}\, s)} \downarrow ((\mathsf{stabilizer}\, s) \times (\mathsf{stabilizer}\, s)) \to (\mathsf{stabilizer}\, s) \wedge \mathsf{e} \in (\mathsf{stabilizer}\, s) \tag{8}$$
$$\Gamma \vDash \mathsf{submonoid}\,(\mathsf{stabilizer}\, s) \tag{9}$$
$$\Gamma \vDash \forall\, s : S \,.\, \mathsf{submonoid}\,(\mathsf{stabilizer}\, s) \tag{10}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from constants always being defined by [20, Axiom A5.2]; (3) follows from Def8 by the Equality Rules [20, Lemma A.13] and Beta-Reduction [20, Axiom A4] applied to (1) and the RHS of the result; (4) follows from (3) and Axiom 4 of T; (5) follows immediately from (4); (6) follows from Thm2, (3), and Axiom 3 of T; (7) follows from (3) and [20, Axiom A5.4]; (8) follows from Def1 by the Equality Rules and Beta-Reduction applied to (7) and the RHS of the result; (9) follows from (4), (5), (6), and (8) by the Tautology Rule [20, Corollary A.46]; (10) follows from (9) by Universal Generalization [20, Theorem A.30] using the fact that s is free in Γ because Γ is a set of sentences. Therefore, $(\star)$ holds. □

5. Thm24 (Thm21-via-MON-ACT-to-MON):
   $\mathsf{MON\text{-}ACTION}(U_{\{M\}}, U_{\{M\}}, \cdot_{(M\times M)\to M}, \mathsf{e}_M, \cdot_{(M\times M)\to M})$
   (first example is a monoid action).

   Proof of the theorem. Similar to the proof of Thm11. □

A.6 Development of ONE-BT-with-SC

1. Thm25 (Thm21-via-MON-ACT-to-ONE-BT-with-SC):
   $$\begin{aligned}\mathsf{MON\text{-}ACTION}(&\mathsf{F}_{\{S\to S\}},\\ &U_{\{S\}},\\ &\circ_{((S\to S)\times(S\to S))\to(S\to S)}|_{\mathsf{F}\times\mathsf{F}},\\ &\mathsf{id}_{S\to S},\\ &\bullet_{((S\to S)\times S)\to S}|_{\mathsf{F}\times S})\end{aligned}$$
   (second example is a monoid action).

   Proof of the theorem. Similar to the proof of Thm11. □

A.7 Development of MON-HOM

1. Thm26:
   $$\begin{aligned}\mathsf{MON\text{-}HOM}(&U_{\{M_1\}},\\ &U_{\{M_2\}},\\ &\cdot_{(M_1\times M_1)\to M_1},\\ &\mathsf{e}_{M_1},\\ &\cdot_{(M_2\times M_2)\to M_2},\\ &\mathsf{e}_{M_2},\\ &\mathsf{h}_{M_1\to M_2})\end{aligned}$$
   (models of MON-HOM define monoid homomorphisms).

Proof of the theorem. Let $T = (L, \Gamma)$ be MON-HOM and $\mathbf{A}_o$ be

$$\mathsf{MON\text{-}HOM}(U_{\{M_1\}}, U_{\{M_2\}}, \cdot_{(M_1 \times M_1) \to M_1}, \mathsf{e}_{M_1}, \cdot_{(M_2 \times M_2) \to M_2}, \mathsf{e}_{M_2}, \mathsf{h}_{M_1 \to M_2}).$$

We must show $(\star)$ $T \models \mathbf{A}_o$.

$$\Gamma \models \mathsf{MONOID}(U_{\{M_1\}}, \cdot_{(M_1 \times M_1) \to M_1}, \mathsf{e}_{M_1}) \tag{1}$$
$$\Gamma \models \mathsf{MONOID}(U_{\{M_2\}}, \cdot_{(M_2 \times M_2) \to M_2}, \mathsf{e}_{M_2}) \tag{2}$$
$$\Gamma \models \mathsf{h}_{M_1 \to M_2} \downarrow U_{\{M_1\}} \to U_{\{M_2\}} \tag{3}$$
$$\Gamma \models \forall\, x, y : U_{\{M_1\}} \,.\, \mathsf{h}\,(x \cdot y) = (\mathsf{h}\, x) \cdot (\mathsf{h}\, y) \tag{4}$$
$$\Gamma \models \mathbf{A}_o \tag{5}$$

(1) and (2) follow similarly to the proof of Thm1; (3) follows from [20, Axiom 5.2] and parts 8 and 9 of Lemma B.1; (4) follows from Axiom 5 of T and part 5 of Lemma B.1; (5) follows from (1)–(4), Axiom 6 of T, and the definition of MON-HOM in Table 10. Therefore, $(\star)$ holds. □

Thm27: $\mathsf{TOTAL}(\mathsf{h}_{M_1 \to M_2})$ ($\mathsf{h}_{M_1 \to M_2}$ is total).

Proof of the theorem. Let $T = (L, \Gamma)$ be MON-HOM.

$$T \models \mathsf{TOTAL}(\mathsf{h}_{M_1 \to M_2})$$

follows from Axiom 5 of T in the same way that $T \models \mathsf{TOTAL}(\cdot_{(M \times M) \to M})$ follows from Axiom 1 of MON as shown in the proof of Thm2. □

2. Thm28 (Thm26-via-MON-HOM-to-MON-4)
   $\mathsf{MON\text{-}HOM}(U_{\{M\}},$
   $U_{\{\{M\}\}},$
   $\cdot_{(M \times M) \to M},$
   $\mathsf{e}_M,$
   $\cdot_{(\{M\} \times \{M\}) \to \{M\}},$
   $\mathsf{E}_{\{M\}},$
   $\mathsf{h}_{M \to \{M\}})$ (example is a monoid homomorphism).

   Proof of the theorem. Similar to the proof of Thm11. □

A.8 Development of MON-over-COF

1. Def9: $\mathsf{prod}_{R \to R \to (R \to M) \to M} =$
   $\mathrm{I}\, f : Z_{\{R\}} \to Z_{\{R\}} \to (Z_{\{R\}} \to M) \to M$.

$$\forall\, m, n : Z_{\{R\}},\, g : Z_{\{R\}} \to M \,.\, f\, m\, n\, g \simeq (m > n \mapsto \mathsf{e} \mid (f\, m\, (n-1)\, g) \cdot (g\, n)) \qquad \text{(iterated product).}$$

Proof that RHS is defined. Let

$$\mathbf{A}_o = \forall\, m, n : Z_{\{R\}},\, g : Z_{\{R\}} \to M \,.\, f\, m\, n\, g \simeq (m > n \mapsto \mathsf{e} \mid (f\, m\, (n-1)\, g) \cdot (g\, n)).$$

Suppose that two functions f_1 and f_2 satisfy $\mathbf{A}_o$. It is easy to see that f_1 and f_2 must be the same function based on the recursive structure of f in $\mathbf{A}_o$. Thus, $\mathbf{A}_o$ specifies a unique function, and so the RHS of Def9 is defined by [20, Axiom A6.1]. □

2. Thm29: $\forall\, m : Z_{\{R\}},\, g : Z_{\{R\}} \to M \,.\, \left(\prod_{i=m}^{m} g\, i\right) \simeq g\, m$ (trivial product).

Proof of the theorem. Let $\mathbf{A}_o$ be the theorem and $T = (L, \Gamma)$ be COM-MON-over-COF extended with Def9. We must show $(\star)$ $T \vDash \mathbf{A}_o$.

Let Δ be the set $\{m \in Z_{\{R\}}, g \in Z_{\{R\}} \to M\}$.

$$\Gamma \cup \Delta \vDash (m : R){\downarrow} \wedge (g : R \to M){\downarrow} \tag{1}$$

$$\Gamma \cup \Delta \vDash \left(\prod_{i=m}^{m} g\, i\right) \simeq \left(\prod_{i=m}^{m-1} g\, i\right) \cdot g\, m \tag{2}$$

$$\Gamma \cup \Delta \vDash \left(\prod_{i=m}^{m-1} g\, i\right) \cdot g\, m \simeq \mathsf{e} \cdot g\, m \tag{3}$$

$$\Gamma \cup \Delta \vDash \mathsf{e} \cdot g\, m \simeq g\, m \tag{4}$$

$$\Gamma \cup \Delta \vDash \left(\prod_{i=m}^{m} g\, i\right) \simeq g\, m \tag{5}$$

$$\Gamma \vDash \mathbf{A}_o \tag{6}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (3) follow from (1) and Def9; (4) follows from Axiom 20 of T; (5) follows from (2), (3), and (4) by the Quasi-Equality Rules [20, Lemma A.4]; and (6) follows from (5) by the Deduction Theorem [20, Theorem A.50] and by Universal Generalization [20, Theorem A.30] using the fact that m and g are not free in Γ since Γ is a set of sentences. Therefore, $(\star)$ holds. □

3. Thm30: $\forall\, m, k, n : Z_{\{R\}},\ g : Z_{\{R\}} \to M$.
$m < k < n \Rightarrow \left(\prod_{i=m}^{k} g\, i\right) \cdot \left(\prod_{i=k+1}^{n} g\, i\right) \simeq \prod_{i=m}^{n} g\, i$
(extended iterated product).

Proof of the theorem. Let $\mathbf{A}_o$ be the theorem and $T = (L, \Gamma)$ be MON-over-COF extended by Def9. We must show (a) $T \vDash \mathbf{A}_o$.

Let Δ be the set

$$\{m \in Z_{\{R\}}, k \in Z_{\{R\}}, n \in Z_{\{R\}}, m < k < n\}.$$

We will prove

$$\text{(b) } \Gamma \cup \Delta \vDash \left(\prod_{i=m}^{k} g\, i\right) \cdot \left(\prod_{i=k+1}^{n} g\, i\right) \simeq \left(\prod_{i=m}^{n} g\, i\right)$$

from all $n > k$ by induction on the n.

Base case: $n = k + 1$. Then:

$$\Gamma \cup \Delta \vDash (m : R)\!\downarrow \wedge\, (k : R)\!\downarrow \wedge\, (n : R)\!\downarrow \wedge\, (g : R \to M)\!\downarrow \tag{1}$$

$$\Gamma \cup \Delta \vDash \left(\prod_{i=m}^{k} g\, i\right) \cdot \left(\prod_{i=k+1}^{n} g\, i\right) \simeq \left(\prod_{i=m}^{k} g\, i\right) \cdot g\, n \tag{2}$$

$$\Gamma \cup \Delta \vDash \left(\prod_{i=m}^{k} g\, i\right) \cdot g\, n \simeq \left(\prod_{i=m}^{n} g\, i\right) \tag{3}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) follows from $n = k + 1$ and Thm29; and (3) follows from $n = k + 1$, (1), and Def9. Thus (b) holds by the Quasi-Equality Rules [20, Lemma A.4] when $n = k + 1$.

Induction step: $n > k + 1$ *and assume*

$$\Gamma \cup \Delta \vDash \left(\prod_{i=m}^{k} g\, i\right) \cdot \left(\prod_{i=k+1}^{n-1} g\, i\right) \simeq \left(\prod_{i=m}^{n-1} g\, i\right).$$

Then:

$$\Gamma \cup \Delta \vDash (m : R)\downarrow \wedge (k : R)\downarrow \wedge (n : R)\downarrow \wedge (g : R \to M)\downarrow \tag{1}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{k} g\,i\Big) \cdot \Big(\prod_{i=k+1}^{n} g\,i\Big) \simeq \Big(\prod_{i=m}^{k} g\,i\Big) \cdot \Big(\Big(\prod_{i=k+1}^{n-1} g\,i\Big) \cdot g\,n\Big) \tag{2}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{k} g\,i\Big) \cdot \Big(\Big(\prod_{i=k+1}^{n-1} g\,i\Big) \cdot g\,n\Big) \simeq \Big(\prod_{i=m}^{n-1} g\,i\Big) \cdot g\,n \tag{3}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n-1} g\,i\Big) \cdot g\,n \simeq \Big(\prod_{i=m}^{n} g\,i\Big) \tag{4}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (4) follows from (1) and Def9; and (3) follows from Axiom 19 of T and the induction hypothesis. Thus (b) holds by the Quasi-Equality Rules [20, Lemma A.4] when $n > k + 1$.

Therefore, (b) holds for all $n > k$, and (a) follows from this by the Deduction Theorem [20, Theorem A.50] and by Universal Generalization [20, Theorem A.30] using the fact that m, k, n, and g are not free in Γ since Γ is a set of sentences. □

A.9 Development of COM-MON-over-COF

1. Thm31: $\forall\, m, n : Z_{\{R\}},\; g, h : Z_{\{R\}} \to M\;.$
   $\Big(\prod_{i=m}^{n} g\,i\Big) \cdot \Big(\prod_{i=m}^{n} h\,i\Big) \simeq \prod_{i=m}^{n} (g\,i) \cdot (h\,i)$
   (product of iterated products).

   Proof of the theorem. Let $\mathbf{A}_o$ be the theorem and $T = (L, \Gamma)$ be COM-MON-over-COF extended by Def9. We must show (a) $T \vDash \mathbf{A}_o$.

   Let Δ be the set $\{n \in Z_{\{R\}}, g \in Z_{\{R\}} \to M\}$. We will prove

   $$\text{(b) } \Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n} g\,i\Big) \cdot \Big(\prod_{i=m}^{n} h\,i\Big) \simeq \prod_{i=m}^{n} (g\,i) \cdot (h\,i)$$

   for all n by induction on the n.

Base case: $n < m$. Then:

$$\Gamma \cup \Delta \vDash (n : R){\downarrow} \wedge (g : R \rightarrow M){\downarrow} \tag{1}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n} g\, i\Big) \cdot \Big(\prod_{i=m}^{n} h\, i\Big) \simeq \mathsf{e} \cdot \mathsf{e} \tag{2}$$

$$\Gamma \cup \Delta \vDash \prod_{i=m}^{n} (g\, i) \cdot (h\, i) \simeq \mathsf{e} \tag{3}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; and (2) and (3) follow from $n < m$, (1), and Def9. Thus (b) holds by Axiom 20 of T and the Quasi-Equality Rules [20, Lemma A.4] when $n < m$.

Induction step: $n \geq m$ and assume

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n-1} g\, i\Big) \cdot \Big(\prod_{i=m}^{n-1} h\, i\Big) \simeq \prod_{i=m}^{n-1} (g\, i) \cdot (h\, i).$$

Then:

$$\Gamma \cup \Delta \vDash (n : R){\downarrow} \wedge (g : R \rightarrow M){\downarrow} \tag{1}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n} g\, i\Big) \cdot \Big(\prod_{i=m}^{n} h\, i\Big) \simeq \Big(\prod_{i=m}^{n-1} g\, i\Big) \cdot g\, n \cdot \Big(\prod_{i=m}^{n-1} h\, i\Big) \cdot h\, n \tag{2}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n-1} g\, i\Big) \cdot g\, n \cdot \Big(\prod_{i=m}^{n-1} h\, i\Big) \cdot h\, n \simeq \Big(\prod_{i=m}^{n-1} g\, i\Big) \cdot \Big(\prod_{i=m}^{n-1} h\, i\Big) \cdot g\, n \cdot h\, n \tag{3}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n-1} g\, i\Big) \cdot \Big(\prod_{i=m}^{n-1} h\, i\Big) \cdot g\, n \cdot h\, n \simeq \Big(\prod_{i=m}^{n-1} (g\, i) \cdot (h\, i)\Big) \cdot (g\, n \cdot h\, n) \tag{4}$$

$$\Gamma \cup \Delta \vDash \Big(\prod_{i=m}^{n-1} (g\, i) \cdot (h\, i)\Big) \cdot (g\, n \cdot h\, n) \simeq \prod_{i=m}^{n} (g\, i) \cdot (h\, i) \tag{5}$$

(1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (5) follow from (1) and Def9; (3) follows from Axiom 21 of T; and (4) follows from the induction hypothesis. Thus (b) holds by the Quasi-Equality Rules [20, Lemma A.4] when $n \geq m$.

Therefore, (b) holds for all n, and (a) follows from this by the Deduction Theorem [20, Theorem A.50] and by Universal Generalization [20, Theorem A.30] using the fact that n and g are not free in Γ since Γ is a set of sentences. □

A.10 Development of COM-MON-ACT-over-COF

1. Thm32: $\forall\, x, y : M,\ s : S \,.\, x \;\mathsf{act}\; (y \;\mathsf{act}\; s) = y \;\mathsf{act}\; (x \;\mathsf{act}\; s)$ (act has commutative-like property).

   Proof of the theorem. Let $\mathbf{A}_o$ be the theorem and $T = (L, \Gamma)$ be COM-MON-ACT-over- COF. We must show ($\star$) $T \models \mathbf{A}_o$.

$$\Gamma \models (x : M)\!\downarrow \wedge\, (y : M)\!\downarrow \wedge\, (s : S)\!\downarrow \tag{1}$$
$$\Gamma \models x \;\mathsf{act}\, (y \;\mathsf{act}\, s) = (x \cdot y) \;\mathsf{act}\, s \tag{2}$$
$$\Gamma \models y \;\mathsf{act}\, (x \;\mathsf{act}\, s) = (y \cdot x) \;\mathsf{act}\, s \tag{3}$$
$$\Gamma \models x \cdot y = y \cdot x \tag{4}$$
$$\Gamma \models y \;\mathsf{act}\, (x \;\mathsf{act}\, s) = (x \cdot y) \;\mathsf{act}\, s \tag{5}$$
$$\Gamma \models x \;\mathsf{act}\, (y \;\mathsf{act}\, s) = y \;\mathsf{act}\, (x \;\mathsf{act}\, s) \tag{6}$$
$$\Gamma \models \mathbf{A}_o \tag{7}$$

   (1) follows from variables always being defined by [20, Axiom A5.1]; (2) and (3) follow from (1) and Axiom 22 of T by Universal Instantiation [20, Theorem A.14]; (4) follows from (1) and Axiom 21 of T by Universal Instantiation; (5) follows from (4) and (3) by Quasi-Equality Substitution [20, Lemma A.2]; (6) follows from (2) and (5) by the Equality Rules [20, Lemma A.13]; (7) follows from (6) by Universal Generalization [20, Theorem A.30] using the fact that x, y, and s are not free in Γ since Γ is a set of sentences. Therefore, ($\star$) holds. □

A.11 Development of STR

1. Def10: $\mathsf{str}_{\{R \to A\}} = [A]$ (string quasitype).

   Proof that RHS is defined. Let T be the top theory of STR-1. We must show ($\star$) $T \models [A]\!\downarrow$. Now $[A]$ stands for

$$\{s : \langle\!\langle A \rangle\!\rangle \mid \exists\, n : \mathbf{C}^{N}_{\{R\}} \,.\, \forall\, m : \mathbf{C}^{N}_{\{R\}} \,.\, (s\; m)\!\downarrow \Leftrightarrow \mathbf{C}_{A \to A \to o} m\; n\}$$

   based on the notational definitions in Table 12. Thus ($\star$) holds because function abstractions are always defined by [20, Axiom A5.11]. □

2. Def11: $\epsilon_{R \rightarrow A} = [\,]_{R \rightarrow A}$ (empty string).

   Proof that RHS is defined. Let T be the top theory of STR-1. We must show $(\star)$ $T \vDash [\,]_{R \rightarrow A}\downarrow$. Now $[\,]_{R \rightarrow A}$ stands for

   $$\lambda\, x : R \,.\, \bot_A$$

   based on the notational definitions in Tables 4 and 12. Thus $(\star)$ holds because function abstractions are always defined by [20, Axiom A5.11]. □

   Def12: $\mathsf{cat}_{((R \rightarrow A) \times (R \rightarrow A)) \rightarrow (R \rightarrow A)} = {+\!+}_{(R \rightarrow A) \rightarrow (R \rightarrow A) \rightarrow (R \rightarrow A)}$ (concatenation).

   Proof that RHS is defined. Let T be the top theory of STR-1. We must show

   $$(\star)\ T \vDash {+\!+}_{(R \rightarrow A) \rightarrow (R \rightarrow A) \rightarrow (R \rightarrow A)}\downarrow.$$

   The pseudoconstant ${+\!+}_{(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta)}$ is defined in Table 12. For all α and β, ${+\!+}_{(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta)}$ denotes the concatenation function for finite sequences over the denotation of β. Therefore, $(\star)$ holds. □

3. Thm33: $\forall\, x : \mathsf{str} \,.\, \epsilon x = x \epsilon = x$ (ϵ is an identity element).

   Proof of the theorem. Let $T = (L, \Gamma)$ be the top theory of STR-1 extended by Def10–Def12. We must show:

   (a) $T \vDash \forall\, x : \mathsf{str} \,.\, \epsilon x = x$.

   (b) $T \vDash \forall\, x : \mathsf{str} \,.\, x \epsilon = x$.

   Let Δ be the set $\{x \in \mathsf{str}\}$. Then:

   $$\Gamma \cup \Delta \vDash \epsilon x = x \tag{1}$$
   $$\Gamma \vDash \forall\, x : \mathsf{str} \,.\, \epsilon x = x \tag{2}$$

   (1) follows from $x \in \mathsf{str}$ and Def12; and (2) follows from (1) by the Deduction Theorem [20, Theorem A.50] and then by Universal Generalization [20, Theorem A.30] using the fact that $(x : R \rightarrow A)$ is not free in Γ since Γ is a set of sentences. Therefore, (a) holds.

   We will prove (c) $\Gamma \cup \Delta \vDash x\epsilon = x$ by induction on the length of x.

   Base case: x is ϵ. Then $\Gamma \cup \Delta \vDash \epsilon\epsilon = \epsilon$ is an instance of (1) above.

Induction step: x is $(a :: y)$ and assume $\Gamma \cup \Delta \vDash y\epsilon = y$. Then:

$$\Gamma \cup \Delta \vDash (a :: y)\epsilon = (a :: y\epsilon) \quad (1)$$
$$\Gamma \cup \Delta \vDash (a :: y\epsilon) = (a :: y) \quad (2)$$

(1) follows from $x \in \mathsf{str}$ and Def12; and (2) follows from the induction hypothesis and (1) by Quasi-Equality Substitution [20, Lemma A.2]. Thus $\Gamma \cup \Delta \vDash (a :: y)\epsilon = (a :: y)$ holds by the Equality Rules [20, Lemma A.13].

Therefore (c) holds, and (b) follows from (c) by the Deduction Theorem [20, Theorem A.50] and then by Universal Generalization [20, Theorem A.30] using the fact that $(x : R \to A)$ is not free in Γ since Γ is a set of sentences. □

4. Thm34: $\forall\, x, y, z : \mathsf{str} \,.\, x(yz) = (xy)z$ (cat is associative).

Proof of the theorem. Let $T = (L, \Gamma)$ be the top theory of STR-1 extended by Def10–Def12. We must show

(a) $T \vDash \forall\, x, y, z : \mathsf{str} \,.\, x(yz) = (xy)z$.

Let Δ be the set $\{x \in \mathsf{str}, y \in \mathsf{str}, z \in \mathsf{str}\}$. We will prove

(b) $\Gamma \cup \Delta \vDash x(yz) = (xy)z$

by induction on the length of x.

Base case: x is ϵ. Then:

$$\Gamma \cup \Delta \vDash \epsilon(yz) = (yz) \quad (1)$$
$$\Gamma \cup \Delta \vDash (yz) = (\epsilon y)z \quad (2)$$

(1) and (2) follow from Thm33. Thus $\Gamma \cup \Delta \vDash \epsilon(yz) = (\epsilon y)z$ holds by the Equality Rules [20, Lemma A.13].

Induction step: x is $(a :: w)$ and assume $\Gamma \cup \Delta \vDash w(yz) = (wy)z$. Then:

$$\Gamma \cup \Delta \vDash (a :: w)(yz) = a :: w(yz) \quad (1)$$
$$\Gamma \cup \Delta \vDash a :: w(yz) = a :: (wy)z \quad (2)$$
$$\Gamma \cup \Delta \vDash a :: (wy)z = (a :: wy)z \quad (3)$$
$$\Gamma \cup \Delta \vDash (a :: wy)z = ((a :: w)y)z \quad (4)$$

(1), (3), and (4) follow from $x \in \mathsf{str}$, $y \in \mathsf{str}$, and $z \in \mathsf{str}$ and Def12; and (2) follows from the induction hypothesis. Thus

$$\Gamma \cup \Delta \vDash (a :: w)(yz) = ((a :: w)y)z$$

holds by the Equality Rules [20, Lemma A.13].

Therefore (b) holds, and (a) follows from (b) by the Deduction Theorem [20, Theorem A.50] and then by Universal Generalization [20, Theorem A.30] using the fact that $(x : R \to A)$, $(y : R \to A)$, and $(z : R \to A)$ are not free in Γ since Γ is a set of sentences. □

5. Thm35 (Thm1-via-MON-over-COF-to-STR-2):
   $\mathsf{MONOID}(\mathsf{str}_{\{R \to A\}}, \mathsf{cat}_{((R \to A) \times (R \to A)) \to (R \to A)}, \epsilon_{R \to A})$
   (strings form a monoid).

   Proof of the theorem. Similar to the proof of Thm11. □

6. Def13 (Def3-via-MON-over-COF-to-STR-2):
   $\mathsf{set\text{-}cat}_{(\{R \to A\} \times \{R \to A\}) \to \{R \to A\}} =$
   $\mathsf{set\text{-}op}_{(((R \to A) \times (R \to A)) \to (R \to A)) \to ((\{R \to A\} \times \{R \to A\}) \to \{R \to A\})}\, \mathsf{cat}$
   (set concatenation).

   Proof that RHS is defined. Similar to the proof that the RHS of Def6 is defined. □

7. Def14 (Def4-via-MON-over-COF-to-STR-2):
   $\mathsf{E}_{\{R \to A\}} = \{\epsilon_{R \to A}\}$ (set identity element).

   Proof that RHS is defined. Similar to the proof that the RHS of Def6 is defined. □

8. Thm36 (Thm12-via-MON-over-COF-1-to-STR-2):
   $\mathsf{MONOID}(\mathcal{P}(\mathsf{str}_{\{R \to A\}}), \mathsf{set\text{-}cat}_{(\{R \to A\} \times \{R \to A\}) \to \{R \to A\}}, \mathsf{E}_{\{R \to A\}})$
   (string sets form a monoid).

   Proof of the theorem. Similar to the proof of Thm11. □

9. Def15 (Def9-via-MON-over-COF-1-to-STR-2):
   $\mathsf{iter\text{-}cat}_{R \to R \to (R \to (R \to A)) \to (R \to A)} =$
   $\mathrm{I}\, f : Z_{\{R\}} \to Z_{\{R\}} \to (Z_{\{R\}} \to (R \to A)) \to (R \to A)\ .$
   $\forall\, m, n : Z_{\{R\}},\, g : Z_{\{R\}} \to (R \to A)\ .\ f\, m\, n\, g \simeq$
   $(m > n \mapsto \epsilon \mid (f\, m\, (n-1)\, g)\ \mathsf{cat}\ (g\, n))$
   (iterated concatenation).

   Proof that RHS is defined. Similar to the proof that the RHS of Def6 is defined. □

B Miscellaneous theorems

Lemma B.1 (Universal Sets)**.** *The following formulas are valid:*

1. $U_{\{\alpha\}}\downarrow$.
2. $U_{\{\alpha\}} \neq \emptyset_{\{\alpha\}}$.
3. $\forall\, x : \alpha \,.\, x \in U_{\{\alpha\}}$.
4. $(\lambda\, \mathbf{x} : \alpha \,.\, \mathbf{B}_\beta) = (\lambda\, \mathbf{x} : U_{\{\alpha\}} \,.\, \mathbf{B}_\beta)$.
5. $(\forall\, \mathbf{x} : \alpha \,.\, \mathbf{B}_o) \Leftrightarrow (\forall\, \mathbf{x} : U_{\{\alpha\}} \,.\, \mathbf{B}_o)$.
6. $(\exists\, \mathbf{x} : \alpha \,.\, \mathbf{B}_o) \Leftrightarrow (\exists\, \mathbf{x} : U_{\{\alpha\}} \,.\, \mathbf{B}_o)$.
7. $(\mathrm{I}\, \mathbf{x} : \alpha \,.\, \mathbf{B}_o) \simeq (\mathrm{I}\, \mathbf{x} : U_{\{\alpha\}} \,.\, \mathbf{B}_o)$.
8. $\mathbf{A}_\alpha\downarrow \Leftrightarrow (\mathbf{A}_\alpha \downarrow U_{\{\alpha\}})$
9. $U_{\{\alpha \to \beta\}} = (U_{\{\alpha\}} \to U_{\{\beta\}})$.
10. $U_{\{\alpha \times \beta\}} = (U_{\{\alpha\}} \times U_{\{\beta\}})$.
11. $U_{\{\{\alpha\}\}} = \mathcal{P}(U_{\{\alpha\}})$.
12. $\boldsymbol{A}_{(\alpha\times\beta)\to\gamma} = \boldsymbol{A}_{(\alpha\times\beta)\to\gamma}|_{U_{\{\alpha\}}\times U_{\{\beta\}}}$.

Proof The proof is left to the reader as an exercise. □

Lemma B.2. *Let T be* MON *extended by the definition* Def2*. The formula*

$$\boldsymbol{A}_M \cdot^{\mathrm{op}} \boldsymbol{B}_M \simeq \boldsymbol{B}_M \cdot \boldsymbol{A}_M$$

is valid in T.

Proof Let $\mathbf{X}_o$ be

$$\mathbf{A}_M \cdot^{\mathrm{op}} \mathbf{B}_M \simeq \mathbf{B}_M \cdot \mathbf{A}_M,$$

N be a model of T, and $\varphi \in \mathsf{assign}(N)$. Suppose that $V_\varphi^N(\mathbf{A}_M)$ or $V_\varphi^N(\mathbf{B}_M)$ is undefined. Then clearly $V_\varphi^N(\mathbf{X}_o) = \mathrm{T}$. Now suppose that $V_\varphi^N(\mathbf{A}_M)$ and $V_\varphi^N(\mathbf{B}_M)$ are defined. Then $V_\varphi^N(\mathbf{X}_o) = \mathrm{T}$ by Def2. □

References

[1] P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth through Proof*. Springer, second edition, 2002.

[2] P. B. Andrews, M. Bishop, S. Issar, D. Nesmith, F. Pfennig, and H. Xi. TPS: A theorem-proving system for classical type theory. *Journal of Automated Reasoning*, 16:321–353, 1996.

[3] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. D. Mosses, D. Sannella, and A. Tarlecki. CASL: the Common Algebraic Specification Language. *Theoretical Computer Science*, 286:153–196, 2002.

[4] S. Autexier, D. Hutter, H. Mantel, and A. Schairer. Towards an evolutionary formal software-development using CASL. In D. Bert, C. Choppy, and P. D. Mosses, editors, *Recent Trends in Algebraic Development Techniques (WADT 1999)*, volume 1827 of *Lecture Notes in Computer Science*, pages 73–88. Springer, 1999.

[5] C. Ballarin. Locales: A module system for mathematical theories. *Journal of Automated Reasoning*, 52:123–153, 2014.

[6] A. Bordg, L. Paulson, and W. Li. Simple type theory is not too simple: Grothendieck's schemes without dependent types. *Experimental Mathematics*, 31:364–382, 2022.

[7] A. Bove, P. Dybjer, and U. Norell. A brief overview of Agda — A functional language with dependent types. In S. Berghofer, T. Nipkow, C. Urban, and M. Wenzel, editors, *Theorem Proving in Higher Order Logics (TPHOLs 2009)*, volume 5674 of *Lecture Notes in Computer Science*, pages 73–78. Springer, 2009.

[8] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[9] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Sasaki, and S. F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, 1986.

[10] L. de Moura, S. Kong, J. Avigad, F. van Doorn, and J. von Raumer. The Lean theorem prover (system description). In A. P. Felty and A. Middeldorp, editors, *Automated Deduction — CADE-25*, volume 9195, pages 378–388, 2015.

[11] Epigram: Marking Dependent Types matter. `http://www.e-pig.org/`. Accessed on 3 January 2025.

[12] F*. `https://www.fstar-lang.org/`. Accessed on 3 January 2025.

[13] W. M. Farmer. A partial functions version of Church's simple theory of types. *Journal of Symbolic Logic*, 55:1269–91, 1990.

[14] W. M. Farmer. A simple type theory with partial functions and subtypes. *Annals of Pure and Applied Logic*, 64:211–240, 1993.

[15] W. M. Farmer. Theory interpretation in simple type theory. In J. Heering, K. Meinke, B. Möller, and T. Nipkow, editors, *Higher-Order Algebra, Logic, and Term Rewriting (HOA 1993)*, volume 816 of *Lecture Notes in Computer Science*, pages 96–123. Springer, 1994.

[16] W. M. Farmer. Formalizing undefinedness arising in calculus. In D. A. Basin and M. Rusinowitch, editors, *Automated Reasoning — IJCAR 2004*, volume 3097 of *Lecture Notes in Computer Science*, pages 475–489. Springer, 2004.

[17] W. M. Farmer. Andrews' type system with undefinedness. In C. Benzmüller, C. E. Brown, J. Siekmann, and R. Statman, editors, *Reasoning in Simple Type Theory: Festschrift in Honor of Peter B. Andrews on his 70th Birthday*, volume 17 of *Studies in Logic*, pages 223–242. College Publications, 2008.

[18] W. M. Farmer. The seven virtues of simple type theory. *Journal of Applied Logic*, 6:267–286, 2008.

[19] W. M. Farmer. LaTeX for Alonzo. `https://imps.mcmaster.ca/doc/latex-for-alonzo.pdf`, 2023 (revised 2024).

[20] W. M. Farmer. *Simple Type Theory: A Practical Logic for Expressing and Reasoning About Mathematical Ideas*. Computer Science Foundations and Applied Logic. Birkhäuser/Springer, second edition, 348 pp., 2025.

[21] W. M. Farmer. Formal mathematics for the masses. In J. Blanchette, Davenport J, P. Koepke, M. Kohlhase, A. Kohlhase, A. Naumowicz, D. Müller, Y. Sharoda, and C. Sacerdoti Coen, editors, *Workshop Papers of the 14th Conference on Intelligent Computer Mathematics (CICM 2021)*, volume 3377 of *CEUR Workshop Proceedings*. CEUR-WS.org, 8 pp., 2023.

[22] W. M. Farmer, J. D. Guttman, and F. J. Thayer. Little theories. In D. Kapur, editor, *Automated Deduction — CADE-11*, volume 607 of *Lecture Notes in Computer Science*, pages 567–581. Springer, 1992.

[23] W. M. Farmer, J. D. Guttman, and F. J. Thayer. IMPS: An Interactive Mathematical Proof System. *Journal of Automated Reasoning*, 11:213–248, 1993.

[24] W. M. Farmer, J. D. Guttman, and F. J. Thayer. *The IMPS 2.0 User's Manual*. The MITRE Corporation, 1998. Available at `https://imps.mcmaster.ca/doc/imps-2.0-manual.pdf`.

[25] Formal Abstracts. `https://formalabstracts.github.io`, 2022. Accessed on 3 January 2025.

[26] F. Garillot. *Generic Proof Tools and Finite Group Theory*. PhD thesis, Ecole Polytechnique X, 2011.

[27] J. Goguen, T. Winkler, J. Meseguer, K. Futatsugi, and J.-P. Jouannaud. Introducing OBJ. In J. Goguen and G. Malcolm, editors, *Software Engineering with OBJ: Algebraic Specification in Action*, volume 2 of *Advances in Formal Methods*, pages 3–167. Springer, 2000.

[28] G. Gonthier, A. Mahboubi, L. Rideau, E. Tassi, and L. Théry. A modular formalisation of finite group theory. In K. Schneider and J. Brandt, editors, *Theorem Proving in Higher Order Logics (TPHOLs 2007)*, volume 4732 of *Lecture Notes in Computer Science*, pages 86–101. Springer, 2007.

[29] M. J. C. Gordon and T. F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, 1993.

[30] T. Hales. Formal abstracts in mathematics. In F. Rabe, W. M. Farmer, G. O. Passmore, and Y. Abdou, editors, *Intelligent Computer Mathematics (CICM 2018)*, volume 11006 of *Lecture Notes in Computer Science*, page xiii. Springer, 2018.

[31] J. Harrison. HOL Light: An overview. In S. Berghofer, T. Nipkow, C. Urban, and M. Wenzel, editors, *Theorem Proving in Higher Order Logics (TPHOLs 2009)*, volume 5674 of *Lecture Notes in Computer Science*, pages 60–66. Springer, 2009.

[32] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[33] M. Iancu. *Towards Flexiformal Mathematics*. PhD thesis, Jacobs University Bremen, 2017.

[34] Idris: A Language for Type-Driven Development. https://www.idris-lang.org/. Accessed on 3 January 2025.

[35] F. Kachapova. Formalizing groups in type theory. *Computing Research Repository (CoRR)*, abs/2102.09125, 2021.

[36] M. Kohlhase. The Flexiformalist Manifesto. In A. Voronkov, V. Negru, T. Ida, T. Jebelean, D. Petcu, S. M. Watt, and D. Zaharie, editors, *14th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012)*, pages 30–35. IEEE Computer Society, 2012.

[37] M. Kohlhase, T. Koprucki, D. Müller, and K. Tabelow. Mathematical models as research data via flexiformal theory graphs. In H. Geuvers, M. England, O. Hasan, F. Rabe, and O. Teschke, editors, *Intelligent Computer Mathematics (CICM 2017)*, volume 10383 of *Lecture Notes in Computer Science*, pages 224–238. Springer, 2017.

[38] M. Kohlhase, F. Rabe, and V. Zholudev. Towards MKM in the large: Modular representation and scalable software architecture. In S. Autexier, J. Calmet, D. Delahaye, P. D. F. Ion, L. Rideau, R. Rioboo, and A. P. Sexton, editors, *Intelligent Computer Mathematics (CICM 2010)*, volume 6167 of *Lecture Notes in Computer Science*, pages 370–384. Springer, 2010.

[39] Metamath Proof Explorer Home Page. http://us.metamath.org/mpeuni/mmset.html. Accessed on 3 January 2025.

[40] T. Mossakowski, C. Maeder, and K. Lüttich. The heterogeneous tool set. In O. Grumberg and M. Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2007)*, volume 4424 of *Lecture Notes in Computer Science*, pages 519–522. Springer, 2007.

[41] R. Nakajima and T. Yuasa, editors. *The IOTA Programming System*, volume 160 of *Lecture Notes in Computer Science*. Springer, 1983.

[42] A. Naumowicz and A. Korniłowicz. A brief overview of Mizar. In S. Berghofer, T. Nipkow, C. Urban, and M. Wenzel, editors, *Theorem Proving in Higher Order Logics (TPHOLs 2009)*, volume 5674 of *Lecture Notes in Computer Science*, pages 67–72. Springer, 2009.

[43] R. P. Nederpelt, J. H. Geuvers, and R. C. De Vrijer, editors. *Selected Papers on Automath*, volume 133 of *Studies in Logic and the Foundations of Mathematics*. North

Holland, 1994.

[44] R. Nickson, O. Traynor, and M. Utting. Cogito Ergo Sum — Providing structured theorem prover support for specification formalisms. In K. Ramamohanarao, editor, *Proceedings of the Nineteenth Australasian Computer Science Conference (ACSC 1996)*, volume 18 of *Australian Computer Science Communications*, pages 149–158, 1996.

[45] U. Norell. *Towards a Practical Programming Language based on Dependent Type Theory*. PhD thesis, Chalmers University of Technology, 2007.

[46] S. Owre, S. Rajan, J. M. Rushby, N. Shankar, and M. Srivas. PVS: Combining specification, proof checking, and model checking. In R. Alur and T. A. Henzinger, editors, *Computer Aided Verification (CAV 1996)*, volume 1102 of *Lecture Notes in Computer Science*, pages 411–414. Springer, 1996.

[47] S. Owre and N. Shankar. Theory interpretations in PVS. Technical Report NASA/CR-2001-211024, NASA, 2001.

[48] L. C. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer, 1994.

[49] L. C. Paulson. Formalising mathematics in simple type theory. In S. Centrone, D. Kant, and D. Sarikaya, editors, *Reflections on the Foundations of Mathematics*, volume 407 of *Synthese Library*, pages 437–453. Springer, 2019.

[50] L. C. Paulson. Large-scale formal proof for the working mathematician — lessons learnt from the ALEXANDRIA Project. *Computing Research Repository (CoRR)*, abs/2305.14407, 2023.

[51] Proof Power. http://www.lemma-one.com/ProofPower/index/index.html. Accessed on 3 January 2025.

[52] F. Rabe and M. Kohlhase. A scalable module system. *Information and Computation*, 230:1–54, 2013.

[53] F. Rabe and C. Schürmann. A practical module system for LF. In J. Cheney and A. Felty, editors, *Proceedings of the Fourth International Workshop on Logical Frameworks and Meta-Languages: Theory and Practice (LFMTP 2009)*, pages 40–48. ACM Press, 2009.

[54] The Rocq Prover. https://rocq-prover.org/. Formerly the Coq Proof Assistant; accessed on 9 May 2025.

[55] J. Rushby, F. von Henke, and S. Owre. An introduction to formal specification and verification using EHDM. Technical Report SRI-CSL-91-02, SRI International, 1991.

[56] D. M. Russinoff. A formalization of finite group theory. *Computing Research Repository (CoRR)*, abs/2205.13347, 2022.

[57] D. Sannella and M. Wirsing. A kernel language for algebraic specification and implementation. In M. Karpinski, editor, *Fundamentals of Computation Theory (FCS 1983)*, volume 158 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 1983.

[58] D. R. Smith. KIDS: A knowledge-based software development system. *IEEE Transactions on Software Engineering*, 16:483–514, 1991.

[59] Y. V. Srinivas and R. Jüllig. Specware: Formal support for composing software. In

B. Möller, editor, *Mathematics of Program Construction (MPC 1995)*, volume 947 of *Lecture Notes in Computer Science*, pages 399–422. Springer, 1995.

[60] X. Yu, A. Nogin, A. Kopylov, and J. Hickey. Formalizing abstract algebra in type theory with dependent records. In *16th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2003)*, Emerging Trends Proceedings, pages 13–27, 2013.

[61] A. Zipperer. *A Formalization of Elementary Group Theory in the Proof Assistant Lean.* PhD thesis, Carnegie Mellon University, 2016.

Received 13 December 2024

Construction Methods of t-norms on Bounded Lattices by t-subnorms

Ting Bao, Hong Jiang, Xiu-juan Hua
School of Mathematics, Xi'an University of Technology, Shaanxi, Xi'an, 710048, China
Corresponding author: X.J. Hua (huaxiujuan@xaut.edu.cn)

Abstract

In this article, we provide methods for constructing t-norms and t-conorms on bounded lattices via t-subnorms and t-superconorms, respectively. These methods have certain conditions, we prove that they are sufficient and necessary, and we use examples to illustrate them. Ultimately, we assert and show that the proposed construction methods are not adaptable to an altered ordinal sum for t-norms and t-conorms through inductive reasoning on bounded lattices.

Keywords: Bounded lattices, T-norms, T-subnorm, Ordinal sum

1 Introduction

Fuzzy logic [33] introduced by Zadeh in 1965, have attracted the attention and interest of a large number of scholars due to their wide application. Typically, fuzzy logic accounts for membership degrees within the unit range $[0, 1]$, yet modern fuzzy logic employs lattices for range. In this reasoning, conjunctions and disjunctions are invariably understood through t-norms and t-conorms.

In 1999, Jenei [15] introduced the concept triangular subnorms (t-subnorms for short) which are generalizations of t-norms. They maintain the property of lying under the minimum. Various properties of t-subnorms have been studied in the literature. Urbanski and Wasowski [28] introduced boundary weak t-norms (i.e., t-subnorm F satisfying $F(1, 1) = 1$) to sum up fuzzy numbers. The study of t-subnorm migrativity, as detailed in [29], specifically outlines the essential and adequate criteria for a t-subnorm with a continuous additive generator to exhibit (α, T)-migrative behavior in relation to any of the three t-norm prototypes T. Zhang et al.[32] revealed the existence of uninorms within Umin (resp. Umax) are characterized by a t-conorm (in that order. t-norm) in conjunction with a t-subnorm (respectively.

t-superconorm), and pioneered unique construction methods utilizing t-subnorms and t-superconorms.

Clifford presented the concept of ordinal sum on a family of semigroups, offering methods to derive novel t-norms from existing ones [4]. Subsequently, Jenei [16] generalized the well-known ordinal sum theorem of semigroups and applied it to construct new t-subnorms and t-norms. Owing to the widespread nature of bounded lattices, numerous scholars have employed ordinal sums to produce novel t-norms and t-conorms in these lattices [1, 5, 12, 22, 25]. Nonetheless, the cumulative ordinal sum of t-norms in a bounded lattice may not invariably result in a t-norm, even with the involvement of a single summand [2].

The literature presents a variety of methods for constructing triangular norms and conorms on bounded lattices [10, 12]. Karaçal and Şanlı [21] explored various techniques to construct t-norms and t-conorms using random elements across bounded lattices. Aşıcı [2] suggested approaches for building t-norms and t-conorms on bounded lattices, utilizing interior and closure operators, in that order. Sun and Liu [27] have done the research about additive generators of t-norms and t-conorms on bounded lattices. The objective of this document is to present two novel methods for developing t-norms and t-conorms by using t-subnorms and t-superconorms on bounded lattices. Although they are all construction methods for t-norms, they all have their own characteristics, providing more selectivity and diversity for research in different fields.

The subsequent sections of this paper are organized as follows. The focus of Section 2 is on exploring core concepts and discoveries relevant to lattices and t-norms in bounded lattices context. The third section presents a pair of novel methods for developing t-norms and t-conorms through the application of t-subnorms and t-superconorms on bounded lattices, accompanied by instances to illustrate these techniques. Progressing further, we utilize multiple examples to illustrate the necessary and sufficient conditions. In the end, it is shown that these are distinct from the t-norms set in [2].

2 Preliminaries

In this section, we recall some important concepts and symbols associated with lattices and t-norms in a bounded lattice. A lattice L is considered bounded when its upper and lower elements are marked as 1 and 0, in that order [3]. In the future, unless stated otherwise, L will be denoted as a bounded lattice.

Let L be a bounded lattice with $f, g \in L\backslash\{0,1\}$, $f \parallel g$ denotes that f is not comparable with g. A subinterval $[f, g]$ of L is defined as $[f, g] = \{x \in L \mid f \leq x \leq$

$g\}$. Similarly, $(f, g] = \{x \in L \mid f < x \leq g\}$, $[f, g) = \{x \in L \mid f \leq x < g\}$ and $(f, g) = \{x \in L \mid f < x < g\}$.

Further, We denote $I_k = \{x \in L \mid x \parallel k\}$, $D_k = [0, k) \times [k, 1] \cup [k, 1] \times [0, k)$ and $D^k = [0, k] \times (k, 1] \cup (k, 1] \times [0, k]$.

Definition 2.1. *[21] An operation $T : L^2 \to L(S : L^2 \to L)$ is said to be a triangular norm (triangular conorm) if it satisfies commutativity, associativity, and increaseness in relation to both variables, possessing a neutral element $e = 1$ ($e = 0$).*

Definition 2.2. *[20] The following are four fundamental t-norms and t-conorms applied to L:*

(i) The largest t-norm $T_\wedge : T_\wedge(x, y) = x \wedge y$.

(ii) The least t-conorm $S_\vee : S_\vee(x, y) = x \vee y$.

(iii) The least t-norm T_D:

$$T_D(x, y) = \begin{cases} x \wedge y & \text{if } \ 1 \in \{x, y\}, \\ 0 & \text{otherwise.} \end{cases}$$

(iv) The largest t-conorm S_D:

$$S_D(x, y) = \begin{cases} x \vee y & \text{if } \ 0 \in \{x, y\}, \\ 1 & \text{otherwise.} \end{cases}$$

Definition 2.3. *[24] A function $J : L^2 \to L$ is called as a t-subnorm when it is commutative, associative, and increasing in relation to both variables, with $J(x, y) \leq x \wedge y$ for every $x, y \in L$.*

Note that each t-norm constitutes a t-subnorm. Conversely, it is not true, in general. Moreover, there are t-subnorms which are not aggregation functions [14]. In fact, each t-subnorm J is increasing and $J(0, 0) = 0$ but $J(1, 1) = 1$ can fail. Nevertheless, creating a t-norm from a t-subnorm is consistently achievable, as illustrated in the following example.

Example 2.4. *[23] If J is a t-subnorm on a bounded lattice L, then T given by*

$$T(x, y) = \begin{cases} J(x, y) & \text{if } \ (x, y) \in (L \setminus \{1\})^2, \\ x \wedge y & \text{otherwise,} \end{cases}$$

is a t-norm on L.

Definition 2.5. *[24] A function $K : L^2 \to L$ is called a t-superconorm when it is commutative, associative, and increasing in relation to both variables, with $x \vee y \leq K(x, y)$ for every $x, y \in L$.*

Example 2.6. *[23] If K is a t-superconorm on a bounded lattice L, then S given by*

$$S(x,y) = \begin{cases} K(x,y) & \text{if } (x,y) \in (L \setminus \{0\})^2, \\ x \vee y & \text{otherwise,} \end{cases}$$

is a t-conorm on L.

Theorem 2.7. *[21] Consider L as a bounded lattice where $k, b \in L$ and $k \neq 0$. Subsequently, the function $T_{k,b} : L \times L \to L$ defined by*

$$T_{k,b}(x,y) = \begin{cases} x \wedge y \wedge b & \text{if } (x \wedge k, y \wedge k) \in [0,k)^2, \\ x \wedge y & \text{otherwise,} \end{cases}$$

is a t-norm.

Theorem 2.8. *[21] Consider L as a bounded lattice where $k, b \in L$ and $k \neq 1$. Subsequently, the function $S_{k,b} : L \times L \to L$ defined by*

$$S_{k,b}(x,y) = \begin{cases} x \vee y \vee b & \text{if } (x \vee k, y \vee k) \in (k,1]^2, \\ x \vee y & \text{otherwise,} \end{cases}$$

is a t-conorm.

Theorem 2.9. *[2] Let L be a bounded lattice with $d \in L$ and $int : L \to L$ be an interior operator such that $x \wedge d = int(x \wedge d)$ for all $x \in I_d$. If V is a t-norm on $[d, 1]$, then the function $T_{int} : L^2 \to L$ defined by*

$$T_{int}(x,y) = \begin{cases} V(x,y) & \text{if } (x,y) \in [d,1)^2, \\ y \wedge d & \text{if } (x,y) \in [d,1) \times I_d, \\ x \wedge d & \text{if } (x,y) \in I_d \times [d,1), \\ x \wedge y \wedge d & \text{if } (x,y) \in I_d \times I_d, \\ x \wedge y & \text{if } x = 1 \text{ or } y = 1, \\ int(x) \wedge int(y) & \text{otherwise,} \end{cases}$$

is a t-norm on L.

Theorem 2.10. *[2] Let L be a bounded lattice with $d \in L$ and and $cl : L \to L$ be a closure operator such that $x \vee d = cl(x \vee d)$ for all $x \in I_d$. If W is a t-conorm on $[0, d]$, then the function $S_{cl} : L^2 \to L$ given by*

$$S_{cl}(x,y) = \begin{cases} W(x,y) & \text{if } (x,y) \in (0,d]^2, \\ y \vee d & \text{if } (x,y) \in (0,d] \times I_d, \\ x \vee d & \text{if } (x,y) \in I_d \times (0,d], \\ x \vee y \vee d & \text{if } (x,y) \in I_d \times I_d, \\ x \vee y & \text{if } x = 0 \text{ or } y = 0, \\ cl(x) \wedge cl(y) & \text{otherwise}, \end{cases}$$

is a t-conorm on L.

3 New construction methods of t-norms by t-subnorms on bounded lattices

In this section, we propose two unique methods for constructing t-norms and t-conorms on bounded lattices, employing t-norms, t-conorms, t-subnorms, and t-superconorms. Furthermore, we present examples to illustrate the unfeasibility of removing constraints on bounded lattices. Additionally, we supply several examples to showcase the distinctiveness of our methods in contrast to those recorded in existing scholarly works.

Theorem 3.1. *Let $(L, \leq, 0, 1)$ be a bounded lattice with $k \in L \setminus \{0, 1\}$. If V is a t-norm on $[k, 1]$ and J is a t-subnorm on L, then the function $T_J : L^2 \to L$ defined by*

$$T_J(x,y) = \begin{cases} V(x,y) & \text{if } (x,y) \in (k,1]^2, \\ x & \text{if } (x,y) \in I_k \times (k,1], \\ y & \text{if } (x,y) \in (k,1] \times I_k, \\ x \wedge y & \text{if } (x,y) \in D^k, \\ J(x,y) & \text{otherwise}, \end{cases}$$

is a t-norm on L iff $s < t$ for all $s \in I_k$, $t \in (k, 1)$.

***Proof*.** **Necessity.** Let the function T_J be a t-norm on L. Suppose that there exit some elements $x \in I_k$, $y \in (k, 1)$ such that $x \parallel y$. Let $z \in (k, 1)$ and $z > x$, then we get $T_J(x, y) = x$ and $T_J(z, y) = V(z, y)$. By the monotonicity of T_J, it should be $x = T_J(x, y) < T_J(z, y) = V(z, y)$. But $V(z, y) \in (k, 1)$, $x \in I_k$, so $x \parallel V(x, y)$, which is a contradiction with $x < V(x, y)$. Thus, $s < t$ for all $s \in I_k$, $t \in (k, 1)$.

Sufficiency. Let $s < t$ for all $s \in I_k$, $t \in (k, 1)$. We prove that the function T_J is a t-norm on L. Obviously, T_J is commutative and $T_J(1, x) = T_J(x, 1) = x$ for all $x \in L$. We need to prove that T_J is monotone and associative. The proof is separated into possible cases as follows.
Monotonicity: We prove that if $x \leq y$ then for all $z \in L$, $T_J(x, z) \leq T_J(y, z)$.

1. Let $x \leq k$.

   1.1. $y \leq k$,

   1.1.1. $z \leq k$ or $z \in I_k$,

$$T_J(x, z) = J(x, z) \leq J(y, z) = T_J(y, z).$$

   1.1.2. $z > k$,

$$T_J(x, z) = x \wedge z = x \leq y = y \wedge z = T_J(y, z).$$

   1.2. $y > k$,

   1.2.1. $z \leq k$,

$$T_J(x, z) = J(x, z) \leq x \wedge z \leq y \wedge z = T_J(y, z).$$

   1.2.2. $z > k$,

$$T_J(x, z) = x \wedge z = x \leq V(y, z) = T_J(y, z).$$

   1.2.3. $z \in I_k$,

$$T_J(x, z) = J(x, z) \leq x \wedge z \leq z = T_J(y, z).$$

   1.3. $y \in I_k$,

   1.3.1. $z \leq k$ or $z \in I_k$,

$$T_J(x, z) = J(x, z) \leq J(y, z) = T_J(y, z).$$

   1.3.2. $z > k$,

$$T_J(x, z) = x \wedge z = x \leq y = T_J(y, z).$$

2. Let $x > k$.

   2.1. $y > k$,

   2.1.1. $z \leq k$,

$$T_J(x, z) = x \wedge z = z = y \wedge z = T_J(y, z).$$

2.1.2. $z > k$,

$$T_J(x,z) = V(x,z) \leq V(y,z) = T_J(y,z).$$

2.1.3. $z \in I_k$,

$$T_J(x,z) = z = T_J(y,z).$$

3. Let $x \in I_k$.

3.1. $y > k$,

3.1.1. $z \leq k$,

$$T_J(x,z) = J(x,z) \leq x \wedge z \leq y \wedge z = T_J(y,z).$$

3.1.2. $z > k$,

$$T_J(x,z) = x \leq V(y,z) = T_J(y,z).$$

3.1.3. $z \in I_k$,

$$T_J(x,z) = J(x,z) \leq x \wedge z \leq z = T_J(y,z).$$

3.2. $y \in I_k$,

3.2.1. $z \leq k$ or $z \in I_k$,

$$T_J(x,z) = J(x,z) \leq J(y,z) = T_J(y,z).$$

3.2.2. $z > k$,

$$T_J(x,z) = x \leq y = T_J(y,z).$$

Associativity: We testify that $T_J(x, T_J(y,z)) = T_J(T_J(x,y),z)$ for all $x,y,z \in L$.

1. Let $x \leq k$.

1.1. $y \leq k$,

1.1.1. $z \leq k$ or $z \in I_k$,

$$T_J(x, T_J(y,z)) = T_J(x, J(y,z)) = J(x, J(y,z)) = J(J(x,y),z).$$

$$= T_J(J(x,y),z) = T_J(T_J(x,y),z).$$

1.1.2. $z > k$,

$$T_J(x, T_J(y,z)) = T_J(x, y \wedge z) = T_J(x,y) = J(x,y).$$

$$= J(x,y) \wedge z = T_J(J(x,y),z) = T_J(T_J(x,y),z).$$

1.2. $y > k$,

1.2.1. $z \leq k$,

$$T_J(x, T_J(y, z)) = T_J(x, y \wedge z) = T_J(x, z) = J(x, z)$$
$$= T_J(x, z) = T_J(x \wedge y, z) = T_J(T_J(x, y), z).$$

1.2.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, V(y, z)) = x \wedge V(y, z) = x = x \wedge z$$
$$= T_J(x, z) = T_J(x \wedge y, z) = T_J(T_J(x, y), z).$$

1.2.3. $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, z) = J(x, z) = T_J(x, z)$$
$$= T_J(x \wedge y, z) = T_J(T_J(x, y), z).$$

1.3. $y \in I_k$,

1.3.1. $z \leq k$ or $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, J(y, z)) = J(x, J(y, z)) = J(J(x, y), z)$$
$$= T_J(J(x, y), z) = T_J(T_J(x, y), z).$$

1.3.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, y) = J(x, y) = J(x, y) \wedge z$$
$$= T_J(J(x, y), z) = T_J(T_J(x, y), z).$$

2. Let $x > k$.

2.1. $y > k$,

2.1.1. $z \leq k$ or $z \in I_k$

$$T_J(x, T_J(y, z)) = T_J(x, J(y, z)) = x \wedge J(y, z) = J(y, z)$$
$$= T_J(y, z) = T_J(x \wedge y, z) = T_J(T_J(x, y), z).$$

2.1.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, y \wedge z) = T_J(x, y) = x \wedge y$$
$$= y = y \wedge z = T_J(y, z) = T_J(T_J(x, y), z).$$

2.2. $y > k$,

2.2.1. $z \leq k$,

$$T_J(x, T_J(y, z)) = T_J(x, y \wedge z) = T_J(x, z) = x \wedge z = z$$

$$= V(x, y) \wedge z = T_J(V(x, y), z) = T_J(T_J(x, y), z).$$

2.2.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, V(y, z)) = V(x, V(y, z)) = V(V(x, y), z)$$

$$= T_J(V(x, y), z) = T_J(T_J(x, y), z).$$

2.2.3. $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, z) = z = T_J(V(x, y), z) = T_J(T_J(x, y), z).$$

2.3. $y > k$,

2.3.1. $z \leq k$,

$$T_J(x, T_J(y, z)) = T_J(x, J(y, z)) = x \wedge J(y, z) = J(y, z)$$

$$= T_J(y, z) = T_J(T_J(x, y), z).$$

2.3.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, y) = y = T_J(y, z) = T_J(T_J(x, y), z).$$

2.3.3. $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, J(y, z)) = J(y, z) = T_J(y, z)$$

$$= T_J(T_J(x, y), z).$$

3. Let $x \leq k$.

3.1. $y \leq k$,

3.1.1. $z \leq k$ or $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, J(y, z)) = J(x, J(y, z)) = J(J(x, y), z)$$

$$= T_J(J(x, y), z) = T_J(T_J(x, y), z).$$

3.1.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, y \wedge z) = T_J(x, y) = J(x, y)$$
$$= J(x, y) \wedge z = T_J(J(x, y), z) = T_J(T_J(x, y), z).$$

3.2. $y > k$,

3.2.1. $z \leq k$,

$$T_J(x, T_J(y, z)) = T_J(x, y \wedge z) = T_J(x, z) = J(x, z)$$
$$= T_J(x, z) = T_J(T_J(x, y), z).$$

3.2.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, V(y, z)) = x = T_J(x, z) = T_J(T_J(x, y), z).$$

3.2.3. $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, z) = J(x, z) = T_J(x, z) = T_J(T_J(x, y), z).$$

3.3. $y \in I_k$,

3.3.1. $z \leq k$ or $z \in I_k$,

$$T_J(x, T_J(y, z)) = T_J(x, J(y, z)) = J(x, J(y, z)) = J(J(x, y), z)$$
$$= T_J(J(x, y), z) = T_J(T_J(x, y), z).$$

3.3.2. $z > k$,

$$T_J(x, T_J(y, z)) = T_J(x, y) = J(x, y) = J(x, y) \wedge z$$
$$= T_J(J(x, y), z) = T_J(T_J(x, y), z).$$

□

In general, the condition $s < t$ for all $s \in I_k$, $t \in (k, 1)$ in Theorem 3.1 cannot be deleted. We use an example to explain it.

Example 3.2. *Let* $(L_1 = \{0, k, e, f, s, g, t, 1\}, \leq, 0, 1)$ *be the bounded lattice depicted in Fig.1, where* $f, s \in I_k$, $e, t \in (k, 1)$ *and* $s \parallel e$, $f \parallel e$. *Let* $V = T_\wedge$ *on* $[k, 1]^2$ *and* $J(x, y) = x \wedge y \wedge k$ *for all* $x, y \in L_1$. *Consequently, the function* T_J *on* L_1*, as depicted in Table 1, does not qualify as a t-norm on* L_1. *Indeed, the function* T_J *fails to meet monotonicity criteria, as evidenced by* $f = T_J(f, e) < T_J(t, e) = V(t, e) = t \wedge e = e$ *and* $s = T_J(s, e) < T_J(t, e) = V(t, e) = t \wedge e = e$*, which are not valid.*

$T_J(x,y)$	0	e	k	f	s	g	t	1
0	0	0	0	0	0	0	0	0
e	0	e	k	f	s	g	e	e
k	0	k	k	0	0	0	k	k
f	0	f	0	0	0	0	f	f
s	0	s	0	0	0	0	s	s
g	0	g	0	0	0	0	g	g
t	0	e	k	f	s	g	t	t
1	0	e	k	f	s	g	t	1

Table 1: The function T_J on L_1 in Example 3.2.

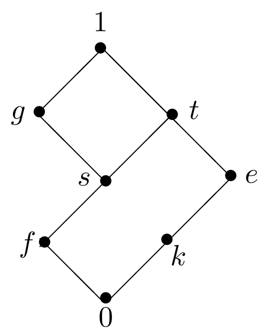


Figure 1: Hasse diagram of the lattice L_1 in Example 3.2.

Corollary 3.3. *Let $(L, \leq, 0, 1)$ be a bounded lattice, $k \in L \setminus \{0,1\}$. If we put $V = T_\wedge$ on $[k,1]^2$ and $J(x,y) = x \wedge y$ for all $x, y \in L, t$. Then the following T_J is an idempotent t-norm on L if and only if $s < t$ for all $s \in I_k$, $t \in (k,1)$.*

$$T_J(x,y) = \begin{cases} x & \text{if } (x,y) \in I_k \times (k,1], \\ y & \text{if } (x,y) \in (k,1] \times I_k, \\ x \wedge y & \text{otherwise.} \end{cases}$$

Observing that T_J differs from $T_{k,b}$ in Theorem 2.7 and T_{int} in Theorem 2.9, an example is provided to substantiate this assertion.

Example 3.4. *Consider the lattice $(L_2 = \{0, k, b, c, d, f, a, m, n, 1\}, \leq, 0, 1)$ be the bounded lattice depicted in Fig.2, where $b, d, f \in I_k$, $c \in (k,1)$ and $b, d, f < c$. Take $V = T_\wedge$ on $[k,1]^2$ and $J(x,y) = x \wedge y \wedge k$ for all $x, y \in L_2$, define $int(0) = 0$, $int(1) = 1$, $int(a) = int(m) = int(n) = a$ and $int(k) = k$. According Theorems 2.7, 2,9 and*

$T_{k,b}(x,y)$	0	k	b	c	d	f	a	m	n	1
0	0	0	0	0	0	0	0	0	0	0
k	0	k	a	k	m	m	a	m	n	k
b	0	a	b	b	b	b	a	a	a	b
c	0	k	b	c	d	f	a	m	n	c
d	0	m	b	d	b	b	a	a	a	d
f	0	m	b	f	b	b	a	a	a	f
a	0	a	a	a	a	a	a	a	a	a
m	0	m	a	m	a	a	a	a	a	m
n	0	n	a	n	a	a	a	a	a	n
1	0	k	b	c	d	f	a	m	n	1

Table 2: The t-norm $T_{k,b}$ on L_2 in Example 3.4.

$T_{int}(x,y)$	0	k	b	c	d	f	a	m	n	1
0	0	0	0	0	0	0	0	0	0	0
k	0	k	a	k	m	m	a	m	n	k
b	0	a	a	a	a	a	a	a	a	b
c	0	k	a	c	m	m	a	a	a	c
d	0	m	a	m	m	m	a	a	a	d
f	0	m	a	m	m	m	a	a	a	f
a	0	a	a	a	a	a	a	a	a	a
m	0	m	a	a	a	a	a	a	a	m
n	0	n	a	a	a	a	a	a	a	n
1	0	k	b	c	d	f	a	m	n	1

Table 3: The t-norm T_{int} on L_2 in Example 3.4.

3.1, we obtain Tables 2,3,4, respectively. It is clear that $T_J(c,f) = T_{k,b}(c,f) = f$, $T_{int}(c,f) = m$, *so* T_J *and* $T_{k,b}$ *are different from* T_{int}, $T_J(b,f) = a$, $T_{k,b}(b,f) = b$, *so* T_J *is different from* $T_{k,b}$. *Therefore,* T_J, $T_{k,b}$ *and* T_{int} *differ from each other.*

We show another construction method of t-conorm by t-superconorm with a constraint as follows.

Theorem 3.5. *Let* $(L, \leq, 0, 1)$ *be a bounded lattice with* $k \in L \setminus \{0,1\}$. *If* W *is a t-conorm on* $[0,k]$ *and* K *is a t-superconorm on* L, *then the function* $S_K : L^2 \to L$ *defined by*

$T_J(x,y)$	0	k	b	c	d	f	a	m	n	1
0	0	0	0	0	0	0	0	0	0	0
k	0	k	a	k	m	m	a	m	n	k
b	0	a	a	b	a	a	a	a	a	b
c	0	k	a	c	d	f	a	m	n	c
d	0	m	a	d	m	m	a	a	a	d
f	0	m	a	f	m	m	a	m	n	f
a	0	a	a	a	a	a	a	a	a	a
m	0	m	a	m	a	m	a	m	n	m
n	0	n	a	n	a	n	a	n	n	n
1	0	k	b	c	d	f	a	m	n	1

Table 4: The t-norm T_J on L_2 in Example 3.4.

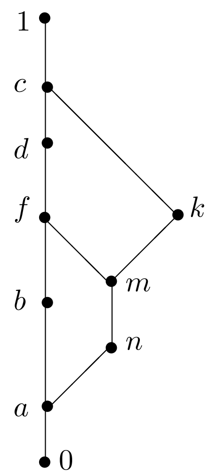


Figure 2: Hasse diagram of the lattice L_2 in Example 3.4.

$$S_K(x,y) = \begin{cases} W(x,y) & \textit{if } (x,y) \in [0,k)^2, \\ x & \textit{if } (x,y) \in I_k \times [0,k), \\ y & \textit{if } (x,y) \in [0,k) \times I_k, \\ x \vee y & \textit{if } (x,y) \in D_k, \\ K(x,y) & \textit{otherwise}, \end{cases}$$

is a t-conorm on L *iff* $u > v$ *for all* $u \in I_k$, $v \in (0,k)$.

$S_K(x,y)$	0	v	k	u	s	t	1
0	0	v	k	u	s	t	1
v	v	v	k	u	s	t	1
k	k	k	k	1	1	k	1
u	u	u	1	1	1	u	1
s	s	s	1	1	1	s	1
t	t	t	k	u	s	t	1
1	1	1	1	1	1	1	1

Table 5: The function S_K on L_3 in Example 3.6.

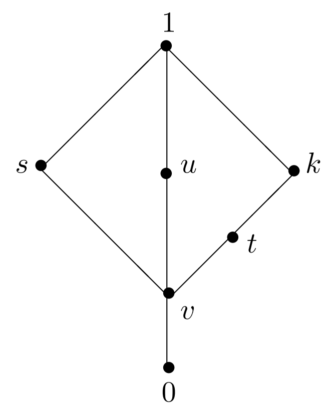


Figure 3: Hasse diagram of the lattice L_3 in Example 3.6.

Proof. The proof process is similar to Theorem 3.1. □

In general, the condition $u > v$ for all $u \in I_k$, $v \in (0,k)$ in Theorem 3.5 cannot be removed. We demonstrate the claim with an example.

Example 3.6. *Let $(L_3 = \{0, v, k, u, s, t, 1\}, \leq, 0, 1)$ be the bounded lattice described by the Hasse diagram in Fig.3, where $u, s \in I_k$, $v, t \in (0,k)$ and $s, u \parallel t$. Let $W = S_\vee$ on $[0,k]^2$ and $K(x,y) = x \vee y \vee t$ for all $x, y \in L_3$. Then the function S_K on L_3, shown in Table 5 is not a t-conorm on L_3. The monotonicity is virtually not satisfied, because $t = S_K(v,t) < S_K(u,t) = u$ and $t = S_K(v,t) < S_K(s,t) = s$ is invalid.*

Corollary 3.7. *Let $(L, \leq, 0, 1)$ be a bounded lattice, $k \in L\backslash\{0,1\}$. If we put $W = S_\vee$ on $[0,k]^2$ and $K(x,y) = x \vee y$ for all $x, y \in L$. Then the following function S_K is*

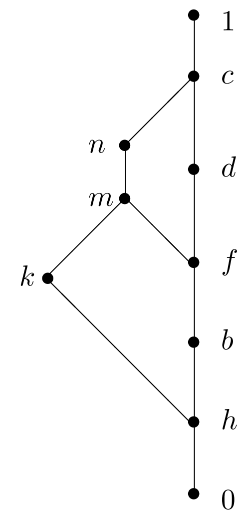


Figure 4: Hasse diagram of the lattice L_4 in Example 3.8.

an idempotent t-conorm on L iff $u > v$ for all $u \in I_k$, $v \in (0,k)$.

$$S_K(x,y) = \begin{cases} x & \text{if } (x,y) \in I_k \times [0,k), \\ y & \text{if } (x,y) \in [0,k) \times I_k, \\ x \vee y & \text{otherwise.} \end{cases}$$

We illustrate that S_K is different from $S_{k,b}$ in Theorem 2.8 and S_{cl} in Theorem 2.10 by using the example below.

Example 3.8. *Consider the lattice $(L_4 = \{0,k,b,c,d,h,f,m,n,1\}, \leq, 0, 1)$ be the bounded lattice described by Fig.4. Take $W = S_\vee$ on $[0,k]^2$ and $K(x,y) = x \vee y \vee d$ for all $x,y \in L_4$ in Theorem 3.5. Define the closure operator $cl : L \to L$ by $cl(0) = 0$, $cl(1) = 1$ and $cl(h) = cl(m) = cl(n) = n$ in Theorem 2.10. From construction methods in Theorem 2.8, 2.10 and 3.5, we obtain $S_K(h,f) = S_{k,b}(h,f) = f$, $S_{cl}(h,f) = m$, so S_K and $S_{k,b}$ are different from S_{cl}, $S_K(m,f) = c$, $S_{k,b}(m,f) = b$, so S_K is different from $S_{k,b}$. Hence, the three t-conorms on the bounded lattice L_4 are different.*

4 The constructions of modified ordinal sum of t-norms and t-conorms on bounded lattices

The scholarly texts contain numerous t-norms, and recursion can yield t-conorms on bounded lattices. As an illustration, Çaylı [5], Dan [10] and Aşıcı [2] introduced methods for analyzing the formation of ordinal sums of t-norms and t-conorms on any bounded lattice. In this section, we explores methods for generating t-norms and t-conorms via t-subnorms and t-superconorms, as retrospectively indicated in Theorem 3.1 and Theorem 3.5, scrutinizing the unfeasibility of forming ordinal sum structures of t-norms and t-conorms on a bounded lattice L through recursive methods.

Proposition 4.1. *Consider $(L, \leq, 0, 1)$ as a bounded lattice and $\{k_0, k_1, k_2, \cdots, k_n\}$ as a finite chain in L, where $1 = k_0 > k_1 > k_2 > \cdots > k_n = 0$. Let $V : [k_1, 1]^2 \to [k_1, 1]$ be a t-norm and J be a t-subnorm on L. It should be indicated that the construction method in Theorem 3.1 cannot be acquired by recursion. Since the function $T_i : [k_i, 1]^2 \to [k_i, 1]$ cannot be obtained as follows, where $x < y$ for all $k \in I_{k_i}$, $y \in [k_i, 1)$, $T_1 = V$ and for $i \in \{2, \cdots, n\}$,*

$$T_i(x,y) = \begin{cases} T_{i-1}(x,y) & \text{if } (x,y) \in [k_{i-1},1]^2, \\ x & \text{if } (x,y) \in I_{k_{i-1}} \times [k_{i-1},1], \\ y & \text{if } (x,y) \in [k_{i-1},1] \times I_{k_{i-1}}, \\ x \wedge y & \text{if } (x,y) \in D_{k_{i-1}}, \\ J(x,y) & \text{otherwise.} \end{cases}$$

An example is provided to demonstrate this claim.

Example 4.2. *Take into account the bounded lattice $L_5 = \{0, k_4, k_3, k_2, k_1, t, s, n, m, p, q, 1\}$ with $0 < k_4 < k_3 < k_2 < k_1 < 1$ described in Fig.5. Choose $J(x,y) = x \wedge y \wedge t$ for all $x, y \in L$ and t-norm $V = T_\wedge : [k_1, 1]^2 \to [k_1, 1]$. Evidently, $x < y$ for all $x \in I_{k_i}$, $y \in [k_i, 1)$. Since $J(m, k_2) = m \wedge k_2 \wedge t = t \notin [k_2, 1]$, the binary operation T_2 cannot be derived on $[k_2, 1]$. Since $J(n, n) = n \wedge n \wedge t = t \notin [k_3, 1]$, the binary operation T_3 cannot be derived on $[k_3, 1]$. Since $J(s, k_4) = s \wedge k_4 \wedge t = t \notin [k_4, 1]$, the binary operation T_4 cannot be derived on $[k_4, 1]$.*

Remark 4.3. *Considering $J(x,y) = x \wedge y$ or $J(x,y) = x \wedge y \wedge k_i$ $(i = 0, 1, 2, \cdots, n)$, where $\{k_0, k_1, k_2, \cdots, k_n\}$ is a finite chain in a bounded lattice L. Then $T_i : [k_i, 1]^2 \to [k_i, 1]$ given in Proposition 4.1 is a t-norm.*

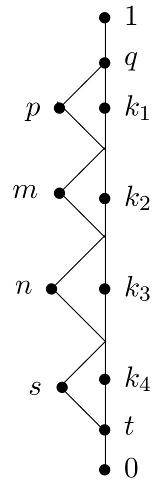


Figure 5: Hasse diagram of the lattice L_5 in Example 4.2.

Proposition 4.4. *Let $(L, \leq, 0, 1)$ be a bounded lattice and $\{k_0, k_1, k_2, \cdots, k_n\}$ be a finite chain in L such that $0 = k_0 < k_1 < k_2 < \cdots < k_n = 1$. Let $W : [0, k_1]^2 \to [0, k_1]$ be a t-conorm and K be a t-superconorm on L. It should be noticed that the construction method in Theorem 3.5 cannot be acquired by recursion. Since the function $S_i : [0, k_i]^2 \to [0, k_i]$ cannot be gained as follows. Where $x > y$ for all $x \in I_{k_i}$, $y \in (0, k_i]$, $S_1 = W$ and for $i \in \{2, \cdots, n\}$,*

$$S_i(x,y) = \begin{cases} S_{i-1}(x,y) & \text{if } (x,y) \in [0,k_{i-1}]^2, \\ x & \text{if } (x,y) \in I_{k_{i-1}} \times [0,k_{i-1}], \\ y & \text{if } (x,y) \in [0,k_{i-1}] \times I_{k_{i-1}}, \\ x \vee y & \text{if } (x,y) \in D^{k_{i-1}}, \\ K(x,y) & \text{otherwise.} \end{cases}$$

Subsequently, the ensuing example serves to demonstrate this deduction.

Example 4.5. *Consider the bounded lattice $L_6 = \{0, k_1, k_2, k_3, k_4, s, t, m, n, p, q, 1\}$ with $0 < k_1 < k_2 < k_3 < k_4 < 1$ depicted by the Hasse diagram in Fig.6. Take $K(x, y) = x \vee y \vee q$ for all $x, y \in L$ and t-conorm $W = S_\vee : [0, k_1]^2 \to [0, k_1]$. It is clear that $x > y$ for all $x \in I_{k_i}$, $y \in (0, k_i]$. Since $K(n, k_2) = n \vee k_2 \vee q = q \notin [0, k_2]$, then the binary operation S_2 on $[0, k_2]$ can not be obtained. Since $K(m, m) = m \vee m \vee q = q \notin [0, k_3]$, the binary operation S_3 on $[0, k_3]$ can not be gained. Since $K(p, k_4) = p \vee k_4 \vee q = q \notin [0, k_4]$, the binary operation S_4 on $[0, k_4]$ can not be achieved, too.*

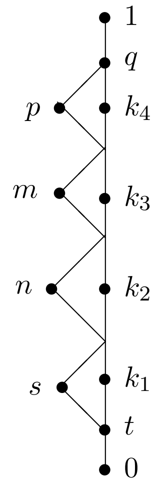


Figure 6: Hasse diagram of the lattice L_6 in Example 4.5.

Remark 4.6. *If we choose $K(x,y) = x \vee y$ or $K(x,y) = x \vee y \vee k_i$ $(i = 0, 1, 2, \cdots, n)$, where $\{k_0, k_1, k_2, \cdots, k_n\}$ is a finite chain in bounded lattice L, then $S_i : [0, k_i]^2 \to [0, k_i]$ given in Proposition 4.4 is a t-conorm.*

5 Concluding remarks

It is a very effective way to obtain construction methods of operators by using t-subnorms and t-superconorms, such as, uninorms and nullnorms. This document delves into and presents innovative approaches for constructing t-norms and t-conorms on a special bounded lattice. Multiple instances were provided to demonstrate the unalterable limitations of construction methods, and we employed examples to highlight the uniqueness of our methods compared to those in the existing literature. In the future, it is challenging to consider how to give definitions of the ordinal sum construction of t-subnorm and t-superconorm on a bounded lattice being a t-norm and t-conorm without any constraints.

Acknowledgments

This research is supported by a grant of National Natural Science Foundation of China (No.12101492) and a grant of Shaanxi Province Science and Technology Plan Project (2024JC-YBMS-070).

References

[1] E. Aşıcı, R. Mesiar, Alternative approaches to obtain t-norms and t-conorms on bounded lattices, Iranian Journal of Fuzzy Systems, 17 (2020) 121-138.

[2] E. Aşıcı, Constructing t-norms and t-conorms by using interior and closure operators on bounded lattices, Iranian Journal of Fuzzy Systems, 19 (2022) 125-140.

[3] G. Birkhoff, Lattice Theory, American Mathematical Society Colloquium Publishers, Providence, 1967.

[4] A. Clifford, Naturally totally ordered commutative semigroups, Am. J. Math. 76 (1954) 631-646.

[5] G.D. Çaylı, On a new class of t-norms and t-conorms on bounded lattices, Fuzzy Sets Syst. 332 (2018) 129-143.

[6] C.A. Drossos, M. Navara, Generalized t-conorms and closure operators, in: Proc. EUFIT'96, Aachen, 1996, pp. 22-26.

[7] B.D.Baets, R. Mesiar, Triangular norm on product lattices, Fuzzy Sets Syst. 104 (1999) 61-75.

[8] D. Dubois, H. prade, Fuzzy sets and systems-Theory and Applications, Academic press. New York, 1980.

[9] D. Dubois, H. prade, A review of fuzzy set aggregation connectives, Inform. Sci. 36 (1985) 85-121.

[10] Y.X. Dan, B.Q. Hu, J.S. Qiao, New construction of t-norms and t-conorms on bounded lattices, Fuzzy Sets Syst. 395 (2020) 40-70.

[11] D. Dubois, H. prade, Fundamentals of Fuzzy Sets, Kluwer Academic Publisher, Boston, 2000.

[12] J. Geng, Z.W. Chen, R.N. Wang, Y.X. Dan, Ordinal sums of triangular norms on sublattices of a bounded lattice, Fuzzy Sets Syst. 462 (2023) 108412.

[13] M. Grabisch, H.T. Nguyen, E.A. Walker, Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference, Kluwer Academic Publishers, Dordrecht, 1995.

[14] M. Grabisch, J.L. Marichal, R. Mesiar, and E. Pap, Aggregation functions, Cambrigde University Press, New York, 2009.

[15] S. Jenei, Fibred triangular norms, Fuzzy Sets Syst. 103 (1999) 67-82.

[16] S. Jenei, A note on the ordinal sum theorem and its consequence for the construction of triangular norms, Fuzzy Sets Syst. 126 (2002) 199-205.

[17] P. Hájek, T. Havránek, R. Jiroušek, Uncertain Information Processing in Expert Systems, CRC Press, Boca Raton, 1992.

[18] W. Homenda, A. Jastrzebska, W. Pedrycz, Multicriteria decision making inspired by human cognitive processes, Appl. Math. Comput. 290 (2016) 392-411.

[19] E.P. Klement, R. Mesiar, E. Pap, Triangular Norms, Kluwer Acad. Publ., Dordrecht, 2000.

[20] F. Karaçal, M.A. İnce, R. Mesiar, Nullnorms on bounded lattices, Inform. Sci. 325 (2015) 227-36.

[21] F. Karaçal, Z. Şanlı, Some new construction methods of t-norms and t-conorms, Fuzzy Sets Syst. 451 (2022) 84-93.

[22] Y. Ouyang, H.P. Zhang, B.D. Baets, Ordinal sums of triangular norms on a bounded lattice, Fuzzy Sets Syst. 408 (2021) 1-12.

[23] E.S. Palmeira, B.C. Bedregal, Extension of t-subnorms on complete Lattices via retractions, 2011 Annual Meeting of the North American Fuzzy Information Processing Society.

[24] E.S. Palmeira, B.C. Bedregal, Extension of fuzzy logic operators defined on bounded lattices via retractions, Comput. Math. Appl. 63 (2012) 1026-1038.

[25] S. Saminger, On ordinal sums of triangular norms on bounded lattices, Fuzzy Sets Syst. 157 (2006) 1403-1416.

[26] B. Schweizer, A. Sklar, Associative functions and statistical triangular inequalities, Publ. Math. 8 (1961) 169-186.

[27] X.R. Sun, H.W. Liu, The additive generators of t-norms and t-conorms on bounded lattices, Fuzzy Sets Syst. 408 (2021) 13-25.

[28] M. Urbaski, J. Wasowski, Fuzzy arithmetic based on boundary weak t-norms, International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, 13 (2005) 27-37.

[29] L.M. Wu, Y. Ouyang, On the migrativity of triangular subnorms. Fuzzy Sets Syst. 226 (2013) 89-98.

[30] R.R. Yager, Aggregation operators and fuzzy systems modelling, Fuzzy Sets Syst. 67 (1994) 129-145.

[31] R.R. Yager, Uninorms in fuzzy system modelling, Fuzzy Sets Syst. 122 (2001) 167-175.

[32] H.P. Zhang, M.X. Wu, Z.D. Wang, Y. Ouyang, B.D. Baets, A characterization of the classes Umin and Umax of uninorms on a bounded lattice, Fuzzy Sets Syst. 423 (2021) 107-121.

[33] L.A. Zadeh, Fuzzy sets, Inform. Control, 8 (1965) 338-353.

 Received 19 February 2025

Pseudo Closures in Quantum B-Algebras

Fengxin Li
School of Science, Xi'an Polytechnic University, Xi'an 710048, P.R. China

Xiaolong Xin *
Business School, Xi'an International University, Xi'an 710077, P.R. China
School of Science, Xi'an Polytechnic University, Xi'an 710048, P.R. China
xlxin@nwu.edu.cn

Xiaofei Yang
School of Science, Xi'an Polytechnic University, Xi'an 710048, P.R. China
yangxiaofei2002@163.com

Abstract

In this paper, we discuss the relationship between quantum B-algebras and pseudo BCI-algebras, and prove that finite normal quantum B-algebras are pseudo BCI-algebras. We introduce a notion of the pseudo closure (denoted as p-closure) for non-empty subsets F of a quantum B-algebra X, denoted by F^{pc}, and explore some of its properties. As an application of p-closure, we characterize the conditions under which a quantum B-algebra becomes a pseudo BCI-algebra and a pseudo BCK-algebra through p-closure. We prove that the p-closure of a subalgebra is still a subalgebra, and the p-closure of a filter is still a filter. Based on the concept of p-closure, we establish the conditions under which a p-closure operator qualifies as a closure operator. Finally, we show that the set of all closed filters of X which $F^{pc} = F$, is a complete lattice.

Keywords: Quantum B-algebra; pseudo BCI-algebra; p-closure; closure operator

*We would like to thank the anonymous referee for his or her valuable comments. Sponsored by a grant of National Natural Science Foundation of China (11971384) and Foreign Expert Program of China (DL2023041002L)

1 Introduction

Birkhoff and Neumann (1936) demonstrated that quantum mechanics exhibits algebraic structures distinct from Boolean algebras, rendering classical probability theory inadequate for fully describing its principles. This led to the development of quantum logic-based probability theory to address quantum events that cannot be observed simultaneously. With the rise of quantum information science (integrating physics, mathematics, computer science, and engineering), probability theories grounded in non-classical logic have become a central research focus. Scholars including Mesiar [6, 7, 20] systematically explored quantum structural theories, marking quantum probability and its applications as a pivotal field in modern interdisciplinary research.

Over recent decades, the extension of algebraic models for non-commutative multiple-valued logics has emerged as a central research focus in the field of fuzzy systems. Notable advancements in this field include non-commutative generalizations of classical algebraic frameworks, such as the non-commutative generalization of MV-algebras by Georgescu and Iorgulescu [14], pseudo effect-algebras defined and investigated by Dvurečenskij and Vetterlein [8, 9], pseudo BL-algebras developed by Di et al. [21], pseudo MTL-algebras proposed by Flondor et al. [12], pseudo BCK-algebras introduced by Georgescu and Iorgulescu [13], pseudo hoops introduced by Georgescu et al. [15], and pseudo BCI-algebras studied by Dudek and Jun [11]. In contemporary research, pseudo BCK-algebras are commonly presented in a dual formulation, characterized by two implication operations (typically denoted as $\to$ and $\rightsquigarrow$) and a greatest element 1. Building upon this framework, pseudo BCI-algebras were introduced in [10] as a generalization unifying pseudo BCK-algebras and BCI-algebras.

Based on the implication operators $\to$ and $\rightsquigarrow$ in quantales, Rump and Yang [26] proposed the concept of quantum B-algebra in 2013 and conducted a series of studies on it. Specifically, quantum B-algebras can provide a unified semantics for non-commutative algebraic logic, covering the majority of implicational algebras such as pseudo BCK/BCI-algebras, (commutative and noncommutative) residuated lattices, pseudo MV/BL/MTL-algebras, pseudo hoop-algebras, and generalized pseudo effect-algebras. This allows us to extend existing conclusions on implicational algebras to quantum B-algebras, making it essential to investigate the relationship between quantum B-algebras and other implicational algebras. Additionally, the properties of quantum B-algebras were investigated in [18, 19, 22, 24, 25].

Closure has played a crucial role in the study of algebraic structures. To extend this concept, some scholars have attempted to generalize the notion of closure, aiming to adapt it to sets that are approximately closed under specific operations, thereby introducing the concept of pseudo-closure. Recently, researchers such as Habib and Moin [1, 16, 17] have proposed the p-closure concept in the contexts of pseudo BCI-algebras and JU-algebras, respectively. Since quantum B-algebras provide a unified semantics for non-commutative algebraic logic, the p-closure defined on pseudo BCI-algebras can be generalized to quantum B-algebras, enriching the algebraic structure of quantum B-algebras.

This paper aims to propose and investigate the concept of p-closure in quantum B-algebras. The paper is organized as follows: Section 2 reviews the fundamental concepts and relevant properties of quantum B-algebras and pseudo BCI-algebras. Section 3 examines the relationship between quantum B-algebras and pseudo BCI-algebras, proving that finite normal quantum B-algebras form pseudo BCI-algebras, and constructs an example of a normal quantum B-algebra that is not a pseudo BCI-algebra. Section 4 introduces the definition of p-closure for non-empty subsets F in a quantum B-algebra X, and explores its related properties. Furthermore, based on the concept of p-closure, a closure operator is defined on the set of all filters of X.

2 Preliminaries

In this section, we review the basic concepts and fundamental properties of quantum B-algebras.

Definition 2.1. *[24] A quantum B-algebra is a poset $(X, \leq)$ with two binary operations $\to$ and $\rightsquigarrow$ satisfying the following conditions, for all $x, y, z \in X$*
(QB_1) $y \to z \leq (x \to y) \to (x \to z)$;
(QB_2) $y \rightsquigarrow z \leq (x \rightsquigarrow y) \rightsquigarrow (x \rightsquigarrow z)$;
(QB_3) $y \leq z$ *implies* $x \to y \leq x \to z$;
(QB_4) $x \leq y \to z$ *iff* $y \leq x \rightsquigarrow z$.

We will refer to $(X, \leq, \to, \rightsquigarrow)$ by its universe X. A quantum B-algebra X is said to be commutative if $x \to y = x \rightsquigarrow y$ for all $x, y \in X$.

Proposition 2.2. *[23, 24] Let $(X, \leq, \to \rightsquigarrow)$ be a quantum B-algebra. Then the following properties hold, for all $x, y, z \in X$,*
(b_1) $x \leq (x \to y) \rightsquigarrow y$, $x \leq (x \rightsquigarrow y) \to y$;
(b_2) $y \leq z$ *implies* $x \rightsquigarrow y \leq x \rightsquigarrow z$;
(b_3) $x \leq y$ *implies* $y \to z \leq x \to z$, $y \rightsquigarrow z \leq x \rightsquigarrow z$;
(b_4) $x \to y = ((x \to y) \rightsquigarrow y) \to y$, $x \rightsquigarrow y = ((x \rightsquigarrow y) \to y) \rightsquigarrow y$;
(b_5) $x \to y \leq (y \to z) \rightsquigarrow (x \to z)$, $x \rightsquigarrow y \leq (y \rightsquigarrow z) \to (x \rightsquigarrow z)$;
(b_6) $x \to (y \rightsquigarrow z) = y \rightsquigarrow (x \to z)$.

A quantum B-algebra X is called unital if there is an element $u \in X$ such that $u \to x = u \rightsquigarrow x = x$ for all $x \in X$. We can prove that the element u is unique [23]. The above element u is said to be a unit element of X. A quantum B-algebra X is called normal if $x \to x = x \rightsquigarrow x = u$, for all $x \in X$, and X is integral if $x \to u = x \rightsquigarrow u = u$, for all $x \in X$.

Lemma 2.3. *[5, 23] Let $(X, \leq, \to, \rightsquigarrow, u)$ be a unital quantum B-algebra. The following hold, for all $x, y \in X$:*
(1) $x \leq y$ iff $u \leq x \to y$ iff $u \leq x \rightsquigarrow y$;
(2) if X is normal, then $x \to u = x \rightsquigarrow u$.

Definition 2.4. *[11] A pseudo BCI-algebra is a structure $(X, \leq, \to, \rightsquigarrow, 1)$, where $\leq$ is a binary relation on X, $\to$ and $\rightsquigarrow$ are binary operations on X and 1 is an element of X, satisfying, for all $x, y, z \in X$,*
$(psBCI_1)$ $x \to y \leq (y \to z) \rightsquigarrow (x \to z)$, $x \rightsquigarrow y \leq (y \rightsquigarrow z) \to (x \rightsquigarrow z)$;
$(psBCI_2)$ $x \leq (x \to y) \rightsquigarrow y$, $x \leq (x \rightsquigarrow y) \to y$;
$(psBCI_3)$ $x \leq x$;
$(psBCI_4)$ if $x \leq y$ and $y \leq x$, then $x = y$;
$(psBCI_5)$ $x \leq y$ iff $x \to y = 1$ iff $x \rightsquigarrow y = 1$.

Theorem 2.5. *[29] Let $(X, \leq, \to, \rightsquigarrow, 1)$ be a pseudo BCI-algebra. Then for all $x, y, z \in X$, we have*
(p_1) $1 \leq x$ implies $x = 1$;
(p_2) $x \leq y$ implies $y \to z \leq x \to z$, $y \rightsquigarrow z \leq x \rightsquigarrow z$;
(p_3) $x \leq y$, $y \leq z$ implies $x \leq z$;
(p_4) $x \to (y \rightsquigarrow z) = y \rightsquigarrow (x \to z)$;
(p_5) $x \leq y \to z$ iff $y \leq x \rightsquigarrow z$;
(p_6) $x \leq y$ implies $z \to y \leq z \to x$, $z \rightsquigarrow y \leq z \rightsquigarrow x$;
(p_7) $x \to y \leq (z \to x) \to (z \to y)$, $x \rightsquigarrow y \leq (z \rightsquigarrow x) \rightsquigarrow (z \rightsquigarrow y)$;
(p_8) $1 \to x = x = 1 \rightsquigarrow x$;
(p_9) $x \to y = ((x \to y) \rightsquigarrow y) \to y$, $x \rightsquigarrow y = ((x \rightsquigarrow y) \to y) \rightsquigarrow y$;
(p_{10}) $x \to y \leq (y \to x) \rightsquigarrow 1$, $x \rightsquigarrow y \leq (y \rightsquigarrow x) \to 1$;
(p_{11}) $(x \to y) \to 1 = (x \to 1) \rightsquigarrow (y \rightsquigarrow 1)$, $(x \rightsquigarrow y) \rightsquigarrow 1 = (x \rightsquigarrow 1) \to (y \to 1)$;
(p_{12}) $x \to 1 = x \rightsquigarrow 1$.

A pseudo BCI-algebra $X = (X, \leq, \to, \rightsquigarrow, 1)$ satisfying $x \leq 1$ for all $x \in X$ is called a pseudo BCK-algebra. Every pseudo BCK-algebra is a pseudo BCI-algebra.

An element a of a quantum B-algebra X is called maximal if for all $x \in X$ the following holds:

$$a \leq x \Rightarrow a = x.$$

Proposition 2.6. *[23, 28] Every pseudo BCI-algebra is a unital quantum B-algebra. And, a quantum B-algebra is a pseudo BCI-algebra if and only if its unit element u is maximal.*

Corollary 2.7. *[26] For a quantum B-algebra X, the following are equivalent:*
(1) *X is a pseudo BCK-algebra;*
(2) *X is integral;*
(3) *X has a greatest element which is the unit element.*

Definition 2.8. *A nonempty subset S is a subalgebra of quantum B-algebras X if it satisfies*

$$x \to y \in S \text{ and } x \rightsquigarrow y \in S \text{ for all } x, y \in S.$$

The study of filters on quantum B-algebras was conducted in [2, 4, 5, 28]. Here we briefly review some key concepts and results.

Definition 2.9. *[4, 5] Let $(X, \leq, \to, \rightsquigarrow, u)$ be a unital quantum B-algebra. A filter of X is a subset $F \subseteq X$ if*
(QF_1) *$u \in F$;*
(QF_2) *$x \in F$, $y \in X$ such that $x \leq y$ implies $y \in F$;*
(QF_3) *$x \in F$, $y \in X$ such that $x \to y \in F$ implies $y \in F$.*

Denote by $\mathcal{F}(X)$ the set of all filters for a unital quantum B-algebra X. If $F \in \mathcal{F}(X)$, (QF_3) can be replaced by: $x \in F$, $y \in X$ such that $x \rightsquigarrow y \in F$ implies $y \in F$, and we have $x \to x \in F$, $x \rightsquigarrow x \in F$ for all $x \in F$. A filter $F \in \mathcal{F}(X)$ is called a closed filter of X if F is closed under $\to$ and $\rightsquigarrow$, i.e., if F is a subalgebra of X. $F \in \mathcal{F}(X)$ is closed if and only if $x \to u \in F$, $x \rightsquigarrow u \in F$, whenever $x \in F$.

Example 2.10. *[5] Let $X = \{0, a, b, c, u, 1\}$ be a poset with $\leq$ defined by $0 \leq a \leq b \leq u \leq 1$ and $a \leq c \leq u$. The Hasse diagram and the operations $\to$ and $\rightsquigarrow$ on X given as follows,*

Table 1: Cayley table for the binary operation "$\to$"

$\to$	0	a	b	c	u	1
0	1	1	1	1	1	1
a	c	u	u	1	1	1
b	c	c	u	c	u	1
c	0	b	b	1	1	1
u	0	a	b	c	u	1
1	0	0	b	0	b	1

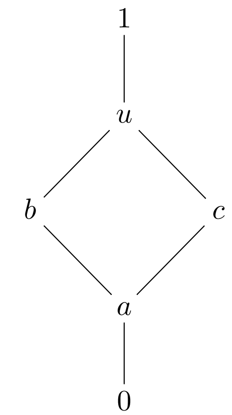


Figure 1: Hasse Diagram of X

Table 2: Cayley table for the binary operation "$\rightsquigarrow$"

$\rightsquigarrow$	0	a	b	c	u	1
0	1	1	1	1	1	1
a	b	u	1	u	1	1
b	0	c	1	c	1	1
c	b	b	b	u	u	1
u	0	a	b	c	u	1
1	0	0	0	c	c	1

Then $(X, \leq, \rightarrow, \rightsquigarrow, u)$ *is a unital quantum B-algebra. We can get the sets* $\{1\}$, $\{0, 1\}$ *and* X *are subalgebras of* X, *and* $\mathcal{F}(X) = \{\{u, 1\}, \{b, u, 1\}, \{c, u, 1\}, X\}$.
(i) *Filters and subalgebras are unrelated. The sets* $\{u, 1\}$ *and* $\{c, u, 1\}$ *are filters of* X *but not subalgebras, and the sets* $\{1\}$ *and* $\{0, 1\}$ *are subalgebras of* X *but not filters;*
(ii) *In Definition 2.9,* (QF_2) *and* (QF_3) *are not equivalent. For instance, consider the set* $A = \{b, c, u, 1\}$. *Since* $u \in A$ *and* $b \rightarrow 0 = c \in A$, *but* $0 \notin A$, *we have that set* A *satisfies* (QF_1) *and* (QF_2), *it fails to satisfy* (QF_3). *Let* $B = \{b, u\}$, *it satisfies* (QF_1) *since* $u \in B$. *And we can check the set* B *satisfies* (QF_3). *Since* $u \leq 1$, $1 \notin B$, *it fails to satisfy* (QF_2).

A mapping $f : A \rightarrow A$ is called a closure operator on a poset $(A, \leq)$ if it satisfies the following conditions for all $x, y \in A$:
(CO_1) $x \leq f(x)$ (extensivity);

(CO_2) $ff(x) = f(x)$ (idempotence);
(CO_3) $x \leq y \Longrightarrow f(x) \leq f(y)$ (isotonicity).

3 The relationship between pseudo BCI-algebras and quantum B-algebras

We have already known that pseudo BCI-algebras and all their classes are quantum B-algebras, although the converse does not hold in general. As demonstrated in Example 2.10, the structure $(X, \leq, \rightarrow, \rightsquigarrow, u)$ is a unital quantum B-algebra yet fails to be a pseudo BCI-algebra.

Assuming that the quantum B-algebras are normal, we discuss the relationship between pseudo BCI-algebras and normal quantum B-algebras.

Proposition 3.1. *Pseudo BCI-algebras are normal quantum B-algebras.*

Proof. According to Proposition 2.6, we know that the pseudo-BCI algebra X is a unital quantum B-algebra. Next, we only need to prove that X is normal. By Theorem 2.5 (p_8), we know that $1 \rightarrow x = x = 1 \rightsquigarrow x$ for all $x \in X$, so 1 is a unit element of X. By Definition 2.4 $(psBCI_3)$ and $(psBCI_5)$, we know $x \rightarrow x = 1 = x \rightsquigarrow x$, so X is normal. Thus, pseudo BCI-algebras are normal quantum B-algebras. □

Example 3.2. *[27] Let* $X = \{0, p, q, r, 1\}$ *together with the order* $0 < p < r < 1, 0 < p < q < 1$. *The binary relations* $\rightarrow$ *and* $\rightsquigarrow$ *as shown below:*

Table 3: Cayley table for the binary operation "→"

$\rightarrow$	0	p	q	r	1
0	1	1	1	1	1
p	p	1	q	1	1
q	0	p	1	1	1
r	0	p	q	1	1
1	0	p	q	r	1

Table 4: Cayley table for the binary operation "⇝"

$\rightsquigarrow$	*0*	*p*	*q*	*r*	*1*
0	*1*	*1*	*1*	*1*	*1*
p	*q*	*1*	*q*	*1*	*1*
q	*0*	*p*	*1*	*1*	*1*
r	*0*	*p*	*q*	*1*	*1*
1	*0*	*p*	*q*	*r*	*1*

It's Hasse diagram is as follows:

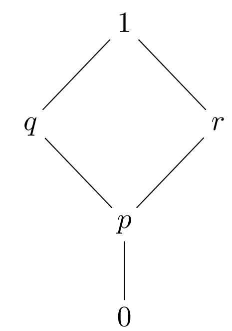


Figure 2: Hasse Diagram of X

Then $(X, \leq, \rightarrow, \rightsquigarrow)$ *is both a pseudo BCI-algebra and a normal quantum B-algebra, as* 1 *is the unit element of* X *and satisfies* $x \rightarrow x = x \rightsquigarrow x = 1$.

For Proposition 3.1, conversely, a normal quantum B-algebra is not necessarily a pseudo BCI-algebra, which can be verified by the following example.

Example 3.3. *Let* $(X, \leq, \rightarrow, \rightsquigarrow)$, *where* $X = \mathbb{R}$, *and* $\mathbb{R}$ *is the set of all real numbers. Define the operations* $\rightarrow$ *and* $\rightsquigarrow$ *:*

$$x \rightarrow y = x \rightsquigarrow y = y - x$$

for all x, $y \in X$. *We have:*
(i) 0 *is the unit element of* X, *since* $0 \rightarrow x = 0 \rightsquigarrow x = x - 0 = x$ *for all* $x \in X$*;*
(ii) X *is a normal quantum B-algebra but not a pseudo BCI-algebra, as it does not satisfy axiom* $(psBCI_5)$.

Proposition 3.4. *Let* X *be a finite normal quantum B-algebra, then* X *is a pseudo BCI-algebra.*

Proof. Since X is finite, every connected component X_i of X has a maximal element and a minimal element. Let a be a minimal element of X_i and u be the unit element. Assume, for contradiction, that u is not a maximal element of X, then there exists an element $b \in X$ such that $u \leq b$. By Proposition 2.2 (b_3), we have $b \to a \leq u \to a = a$. Since a is a minimal element of X, it follows that $b \to a = a$. Therefore, $a \leq b \to a$. By Definition 2.1 (QB_4), this implies $b \leq a \rightsquigarrow a = u$. Hence $b = u$, which means the unit element u is a maximal element of X_i. By Proposition 2.6, X is a pseudo BCI-algebra. □

Example 3.5. *[3] Consider the set $X = \{a, b, c, d, 1\}$ with the partial order $\leq$ defined by $b \leq a \leq 1$, $d \leq c$. Define the operations $\to$ and $\rightsquigarrow$ on X by the tables below:*

Table 5: Cayley table for the binary operation "$\to$"

$\to$	a	b	c	d	1
a	1	b	c	d	1
b	1	1	c	c	1
c	c	c	1	a	c
d	c	c	1	1	c
1	a	b	c	d	1

Table 6: Cayley table for the binary operation "$\rightsquigarrow$"

$\rightsquigarrow$	a	b	c	d	1
a	1	b	c	d	1
b	1	1	c	c	1
c	c	c	1	a	c
d	c	c	1	1	c
1	a	b	c	d	1

One can check that $(X, \leq, \to, \rightsquigarrow, 1)$ is a finite normal quantum B-algebra. Moreover, X is in fact a BCI-algebra.

According to Propositions 3.1 and 3.4, we know that every pseudo BCI-algebra is a normal quantum B-algebra, but the converse does not hold in general. A counterexample is given in Example 3.3. However, the converse does hold for finite normal quantum B-algebras.

4 P-closure with respect to filters

In Section 3, we have proved that finite normal quantum B-algebras are pseudo BCI-algebras. In [16] and [17], the authors discuss p-closure in pseudo BCI-algebras. Now we introduce the concept of p-closure into quantum B-algebras and investigate some related properties.

Definition 4.1. *For any non-empty subset F of a quantum B-algebra X, we define the p-closure of F by the set*

$$F^{pc} = \{x \in X | x \to a \in F \text{ and } x \rightsquigarrow a \in F \text{ for some } a \in F\}.$$

Obviously, if X is unital, then $u \in F^{pc}$.

Lemma 4.2. *For any non-empty subsets E and F of a normal quantum B-algebra X, the following hold:*
(1) if $F \subseteq E$, then $F^{pc} \subseteq E^{pc}$;
(2) if $u \in F$, then $F \subseteq F^{pc}$.

Proof. (1) Let $x \in F^{pc}$. By Definition 4.1, there exists an element $a \in F \subseteq E$ such that $x \to a \in F \subseteq E$ and $x \rightsquigarrow a \in F \subseteq E$. Hence, $x \in E^{pc}$, which implies $F^{pc} \subseteq E^{pc}$.
(2) If $u \in F$, then $x \to x = x \rightsquigarrow x = u \in F$ for all $x \in F$. Thus, $x \in F^{pc}$. Therefore $F \subseteq F^{pc}$. □

Example 4.3. *Consider the structure $(X, \leq, \to, \rightsquigarrow, 0)$ from Example 3.3. Let $F = \{x \in X | -1 \leq x \leq 1\}$, and $E = \{x \in X | -2 \leq x \leq 2\}$. We can get $F \subseteq E$ and $0 \in F$. From the definition of p-closure, it follows that $F^{pc} = \{x \in X | -2 \leq x \leq 2\}$ and $E^{pc} = \{x \in X | -4 \leq x \leq 4\}$, then $F \subseteq F^{pc}$, $E \subseteq E^{pc}$ and $F^{pc} \subseteq E^{pc}$.*

In the following example, we show that the condition $u \in F$ in Lemma 4.2(2) is indispensable in general.

Example 4.4. *Consider the structure $(X, \leq, \to, \rightsquigarrow, 0)$ from Example 3.3, let $F = \{1, 2, 3\}$, and $0 \notin F$. We can get $F^{pc} = \{-1, -2, 0, 1, 2\}$, then $F \nsubseteq F^{pc}$.*

For any normal quantum B-algebras X the set

$$K(X) = \{x \in X | x \leq u\},$$

$$K^{+}(X) = \{x \in X | x \geq u\},$$

we have the following conclusion:
(i) $K(X)$ and $K^{+}(X)$ are not subalgebras of X. For Example 3.3, $K(X) = \{x \in$

$X|x \leq 0\}$ and $K^+(X) = \{x \in X|x \geq 0\}$ are not subalgebras of $\mathbb{R}$.
(ii) For any $x \in K(X)$, $y \in K^+(X)$, we get $x \to y \in K^+(X)$ and $x \rightsquigarrow y \in K^+(X)$. Indeed, if $x \leq u$ and $y \in K^+(X)$, then $u \leq y \leq x \to y$ and $u \leq y \leq x \rightsquigarrow y$.
(iii) For any $x \in K^+(X)$, $y \in K(X)$, we get $x \to y \in K(X)$ and $x \rightsquigarrow y \in K(X)$. Since if $x \in K^+(X)$, $y \in K(X)$, we have $x \to y \leq y \leq u$ and $x \rightsquigarrow y \leq y \leq u$.

Proposition 4.5. *Let X be a normal quantum B-algebra. Then, an element a is maximal element of X iff $\{a\}^{pc} = K(X)$.*

Proof. ($\Rightarrow$) Let a be a maximal element of X. If $x \in \{a\}^{pc}$, we have $x \to a = x \rightsquigarrow a = a$ for $a \in \{a\}$. By Proposition 2.2 (b_6), we get $u = a \rightsquigarrow a = a \rightsquigarrow (x \to a) = x \to (a \rightsquigarrow a) = x \to u$. Then $u \leq x \to u$, according to Lemma 2.3 (1), we have $x \leq u$. It follows that $x \in K(X)$. Hence, $\{a\}^{pc} \subseteq K(X)$. On the other hand, if $x \in K(X)$, then $x \leq u$, so that $a = u \to a \leq x \to a$, for all $x \in X$. Since a is a maximal of X, we get $x \to a = a$. Similarly, $x \rightsquigarrow a = a$. It follows that $x \in \{a\}^{pc}$. Thus, $\{a\}^{pc} = K(X)$.
($\Leftarrow$) If $\{a\}^{pc} = K(X)$. Let $b \in X$ with $a \leq b$. Then $b \to a \leq a \to a = u$, so that $b \to a \in K(X) = \{a\}^{pc}$. Thus, $(b \to a) \rightsquigarrow a = a$, and so $b \to a = b \to ((b \to a) \rightsquigarrow a) = (b \to a) \rightsquigarrow (b \to a) = u$. Then $u \leq b \to a$. We get $b \leq a$ by Lemma 2.3 (1). Then a is a maximal element of X. □

In the following theorem, we give some equivalent conditions for a normal quantum B-algebra becoming a pseudo BCI-algebra or a pseudo BCK-algebra.

Theorem 4.6. *Let X be a normal quantum B-algebra. Then the following are equivalent:*
(a) X is a pseudo BCI-algebra;
(b) $\{u\}^{pc} = K(X)$;
(c) u is maximal.

Proof. Using Proposition 2.6 and Proposition 4.5, the proof is straightforward. □

Theorem 4.7. *Let X be a normal quantum B-algebra. Then the following are equivalent:*
(a) X is a pseudo BCK-algebra;
(b) $\{u\}^{pc} = X$;
(c) $F^{pc} = X$ for any subset F of X containing u.

Proof. $(a) \Rightarrow (b)$ Suppose X is a pseudo BCK-algebra. By Corollary 2.7, we have $x \to u = x \rightsquigarrow u = u \in \{u\}$ for all $x \in X$, then $x \in \{u\}^{pc}$. Thus, $\{u\}^{pc} = X$.
$(b) \Rightarrow (a)$ Let $\{u\}^{pc} = X$. For all $x \in X$, we get $x \to u = u = x \rightsquigarrow u$, then X is

an integral quantum B-algebra. According to Corollary 2.7, we get X is a pseudo BCK-algebra.
$(a) \Rightarrow (c)$ If X is a pseudo BCK-algebra, $F^{pc} \subseteq X$ is obvious. Moreover, $x \to u = x \rightsquigarrow u = u \in F$ for all $x \in X$, hence $x \in F^{pc}$. We conclude that $F^{pc} = X$.
$(c) \Rightarrow (a)$ Let $F^{pc} = X$. In particular, take $F = \{u\}$, then $\{u\}^{pc} = X$. From the equivalence of (a) and (b), we get X is a pseudo BCK-algebra. □

Example 4.8. (i) *For Example 3.5, X is a pseudo BCI-algebra, and $u = 1$ is a maximal element of X. Then $\{u\}^{pc} = \{1\}^{pc} = \{a, b, 1\} = K(X)$.*
(ii) *For Example 3.2, X is a pseudo BCK-algebra, and $u = 1$ is the greatest element of X. Then $\{u\}^{pc} = \{1\}^{pc} = \{0, p, q, r, 1\} = X$, and $F^{pc} = X$ for any subset F of X containing 1.*

Assuming that the normal quantum B-algebra is commutative, we obtain the following results.

Lemma 4.9. *Let F be a subalgebra of a normal quantum B-algebra X. Then the following statements are equivalent: for all $x \in X$,*
(1) $x \in F^{pc}$;
(2) $x \to u \in F$;
(3) $x \to u \in F^{pc}$.

Proof. (1) $\Rightarrow$ (2) Let $x \in F^{pc}$. By Definition 4.1, there exists $a \in F$ such that $x \to a \in F$, and so, $a \rightsquigarrow (x \to a) \in F$. We get $a \rightsquigarrow (x \to a) = x \to (a \rightsquigarrow a) = x \to u$ from Proposition 2.2 (b_6). Therefore, $x \to u \in F$.
(2) $\Rightarrow$ (3) This is obvious by Lemma 4.2 (2).
(3) $\Rightarrow$ (1) Let $x \to u \in F^{pc}$. Then there exists $a \in F$ such that $(x \to u) \to a \in F$ and $(x \to u) \rightsquigarrow a \in F$. Since F is a subalgebra, we obtain $a \rightsquigarrow ((x \to u) \to a) \in F$. By using Proposition 2.2 (b_6), we have $a \rightsquigarrow ((x \to u) \to a) = (x \to u) \to (a \rightsquigarrow a) = (x \to u) \to u \in F$, then $x \rightsquigarrow ((x \to u) \to u) = (x \to u) \rightsquigarrow (x \to u) = u \in F$. Since X is commutative, $x \to ((x \to u) \to u) \in F$. Thus, $x \in F^{pc}$. □

Theorem 4.10. *If F is a subalgebra of a normal quantum B-algebra X, then F^{pc} is a subalgebra of X containing F.*

Proof. Suppose F is a subalgebra of X. By Lemma 4.2 (2), $F \subseteq F^{pc}$ is obvious. Now we just need to prove that F^{pc} is a subalgebra of X. Let $x, y \in F^{pc}$. By Lemma 4.9, we get $x \to u \in F$ and $y \to u \in F$. Since X is commutative, and by Proposition

2.2 (b_6), we have

$$\begin{aligned}(x \to y) \rightsquigarrow (x \to u) &= x \to ((x \to y) \rightsquigarrow u)\\ &= x \to ((x \to y) \rightsquigarrow (y \to y))\\ &= x \to (y \to ((x \to y) \rightsquigarrow y))\\ &= x \to (y \rightsquigarrow ((x \to y) \rightsquigarrow y))\\ &= y \rightsquigarrow (x \to ((x \to y) \rightsquigarrow y))\\ &= y \rightsquigarrow ((x \to y) \rightsquigarrow (x \to y))\\ &= y \rightsquigarrow u\\ &= y \to u.\end{aligned}$$

Hence, $(x \to y) \rightsquigarrow (x \to u) = y \to u \in F$, and so, $(x \to y) \to (x \to u) \in F$. Thus, $x \to y \in F^{pc}$. Similarly, we have $x \rightsquigarrow y \in F^{pc}$. Therefore, F^{pc} is a subalgebra of X. □

In Theorem 4.10, the condition that F is a subalgebra of X is a sufficient condition but not a necessary condition, and the converse does not necessarily hold. Below, we use a specific example to show this result.

Example 4.11. *The structure $(X, \leq, \to, \rightsquigarrow, 0)$ in Example 3.3 is commutative.*
(1) Let $F = \{2n|n \in \mathbb{Z}\}$, we can easily verify that F is a subalgebra of X. By the definition of F^{pc}, we have $F^{pc} = \{2n|n \in \mathbb{Z}\} = F$. Then F^{pc} is also a subalgebra of X.
(2) Let $F = \{2n+1|n \in \mathbb{Z}\}$, we get $F^{pc} = \{2n|n \in \mathbb{Z}\}$, where F^{pc} is a subalgebra of X, but F is not.

Considering that F is a filter for the normal quantum B-algebra X, we obtain the following results.

Theorem 4.12. *If F is a filter of a normal quantum B-algebra X, then so is F^{pc}.*

Proof. (1) Clearly, $u \in F$ from F is a filter. Using Lemma 4.2 (2), we get $F \subseteq F^{pc}$, and so $u \in F^{pc}$.
(2) Let's prove that $x \in F^{pc}$, $y \in X$ such that $x \to y \in F^{pc}$ implies $y \in F^{pc}$. If $x \to y \in F^{pc}$ for any $x \in F^{pc}$, $y \in X$. Thus, there exists $a, b \in F$ such that $x \to a \in F$, $x \rightsquigarrow a \in F$, and $(x \to y) \to b \in F$, $(x \to y) \rightsquigarrow b \in F$. Now, since $a \leq (a \to u) \rightsquigarrow u = (a \to u) \rightsquigarrow (b \to b) = b \to ((a \to u) \rightsquigarrow b)$, we get $b \to ((a \to u) \rightsquigarrow b) \in F$, and so, $(a \to u) \rightsquigarrow b \in F$. Using Definition 2.1 ($QB_1$) and

Properties 2.2, we have

$$\begin{aligned}u &= y \rightsquigarrow y\\ &= y \rightsquigarrow (u \to y)\\ &\le y \rightsquigarrow ((a \to u) \to (a \to y))\\ &= y \rightsquigarrow ((a \to u) \rightsquigarrow (a \to y))\\ &= y \rightsquigarrow (a \to ((a \to u) \rightsquigarrow y))\\ &= a \to (y \rightsquigarrow ((a \to u) \rightsquigarrow y))\\ &\le (x \to a) \to (x \to (y \rightsquigarrow ((a \to u) \rightsquigarrow y))).\end{aligned}$$

Since $u \in F$, we get $(x \to a) \to (x \to (y \rightsquigarrow ((a \to u) \rightsquigarrow y))) \in F$, and hence $x \to (y \rightsquigarrow ((a \to u) \rightsquigarrow y)) \in F$ since $x \to a \in F$. On the other hand, using Properties 2.2, we have

$$\begin{aligned}x \to (y \rightsquigarrow ((a \to u) \rightsquigarrow y)) &= y \rightsquigarrow (x \to ((a \to u) \rightsquigarrow y))\\ &= y \rightsquigarrow ((a \to u) \rightsquigarrow (x \to y))\\ &\le y \rightsquigarrow (((x \to y) \rightsquigarrow b) \to ((a \to u) \rightsquigarrow b))\\ &= ((x \to y) \rightsquigarrow b) \to (y \rightsquigarrow ((a \to u) \rightsquigarrow b))\end{aligned}$$

This implies that $y \rightsquigarrow ((a \to u) \rightsquigarrow b) \in F$ since $x \to (y \rightsquigarrow ((a \to u) \rightsquigarrow y)) \in F$ and $(x \to y) \rightsquigarrow b \in F$. Since X is commutative, $y \to ((a \to u) \rightsquigarrow b) \in F$. Therefore $y \in F^{pc}$.
(3) Finally, we need to prove: $x \in F^{pc}$, $y \in X$ such that $x \le y$ implies $y \in F^{pc}$.
Let $x \le y$ for any $x \in F^{pc}$, $y \in X$. Then, by Lemma 2.3 (1) and Lemma 4.2 (2), we get $u \le x \to y$, and so by $u \in F$, we get $x \to y \in F \subseteq F^{pc}$. According to (2) above, we have $y \in F^{pc}$. By Definition 2.9, we conclude that F^{pc} is a filter of X. □

Theorem 4.13. *If F is a filter of a normal quantum B-algebra X, then F^{pc} is a closed filter of X.*

Proof. If F is a filter of normal quantum B-algebra X, then F^{pc} is also a filter of X, as proven in Theorem 4.12. To show F^{pc} is closed, we prove that $x \to u \in F^{pc}$, $x \rightsquigarrow u \in F^{pc}$ for all $x \in F^{pc}$. Let $x \in F^{pc}$. Then $x \to a \in F$ and $x \rightsquigarrow a \in F$ for some $a \in F$. By Proposition 2.2 (b_1) and (b_6), we get $a \le (a \to u) \rightsquigarrow u = (a \to u) \rightsquigarrow (a \to a) = a \to ((a \to u) \rightsquigarrow a)$. Then $(a \to u) \rightsquigarrow a \in F$ and $(a \to u) \to a \in F$ since $a \in F$. Therefore, $a \to u \in F^{pc}$. By Definition 2.1 (QB_1), we get $a \to u \le (x \to a) \to (x \to u)$, then $(x \to a) \to (x \to u) \in F^{pc}$ since F^{pc} is a filter of X. Thus, $x \to u \in F^{pc}$ because $x \to a \in F \subseteq F^{pc}$. Since X is commutative, $x \rightsquigarrow u \in F^{pc}$. Therefore, F^{pc} is a closed filter of X. □

From Theorems 4.12 and 4.13, we see that the p-closure of a filter is also a filter and is closed. Now, we consider whether a set whose p-closure is a filter is itself a filter.

Example 4.14. *Consider the structure* $(X, \leq, \to, \rightsquigarrow, 0)$ *from Example 3.3. Let* $F = \{x \in X | 0 \leq x\}$*. Obviously,* F *is a filter of* X*, and from the definition of* F^{pc}*, we get that* $F^{pc} = \mathbb{R}$ *is also a filter of* X*, and is a closed filter. Conversely, if* $F = \{x \in X | x \leq 0\}$ *is taken,* $F^{pc} = \mathbb{R}$ *is a closed filter, but* F *is not a filter of* X*.*

In the following theorem, we give some equivalent conditions for a filter to be closed on normal quantum B-algebras.

For any non-empty subset F of a normal a quantum B-algebra X, we denote $F_u = \{x \in F | x \to u \in F\}$.

Theorem 4.15. *Let* F *be a filter of a normal quantum B-algebra* X*. Then the following are equivalent:*
(1) F *is closed;*
(2) $F = F_u$*;*
(3) $F^{pc} = F_u^{pc}$*.*

Proof. (1) $\Rightarrow$ (2) Assume that filter F is closed. The statement $F_u \subseteq F$ is obviously true. Now, we show that $F \subseteq F_u$. Let $x \in F$, we have $x \to u \in F$ by $u \in F$, then $x \in F_u$. Thus, $F \subseteq F_u$. Therefore, $F = F_u$.
(2) $\Rightarrow$ (3) This is obvious.
(3) $\Rightarrow$ (1) Let $F^{pc} = F_u^{pc}$. For all $x \in F$, we have $x \in F^{pc} = F_u^{pc}$. By Theorem 4.13, we have F^{pc} and F_u^{pc} are closed filters of X, thus $u \in F_u^{pc}$ and $x \to u \in F_u^{pc}$. Then there exists $a \in F_u \subseteq F$ such that $(x \to u) \to a \in F_u$ and $(x \to u) \rightsquigarrow a \in F_u$. From the definition of F_u, it follows that $((x \to u) \rightsquigarrow a) \to u \in F$. By Proposition 2.2, we have

$$\begin{aligned} a &= u \to a \\ &\leq (x \to u) \to (x \to a) \\ &= (x \to u) \rightsquigarrow (x \to a) \\ &= x \to ((x \to u) \rightsquigarrow a) \\ &\leq (((x \to u) \rightsquigarrow a) \to u) \rightsquigarrow (x \to u). \end{aligned}$$

Hence, by $a \in F$ and $((x \to u) \rightsquigarrow a) \to u \in F$, we have $x \to u \in F$. By commutativity, $x \rightsquigarrow u \in F$. Therefore, F is closed. ☐

Example 4.16. *Consider the structure* $(X, \leq, \to, \rightsquigarrow, 0)$ *from Example 3.3. Let* $F = \mathbb{R}$*, where* F *is the filter of* X *and is closed, we can obtain* $F_0 = \mathbb{R}$ *and* $F^{pc} = \mathbb{R} = F_0^{pc}$*,* *so $F = F_0$ and $F^{pc} = F_0^{pc}$. If we let $F = \{x \in X|0 \leq x\}$, F is a filter but not closed, we get $F_0 = \{x \in X|x \leq 0\}$ and $F^{pc} = \mathbb{R} = F_0^{pc}$, but $F \neq F_u$.*

Next, we attempt to prove the idempotence of the p-closure.

Theorem 4.17. *Let F be a filter of a normal quantum B-algebra X. Then $F^{pc} = (F^{pc})^{pc}$.*

Proof. By Theorem 4.12 and Lemma 4.2 (2), $F^{pc} \subseteq (F^{pc})^{pc}$ is immediate. On the other hand, let $x \in (F^{pc})^{pc}$. According to Theorem 4.13, it follows that F^{pc} is a closed filter of X. Thus, $x \to u \in F^{pc}$. Then $(x \to u) \to a \in F$ and $(x \to u) \rightsquigarrow a \in F$ for some $a \in F$. Using Proposition 2.2 (b_1) and (b_6), we get

$$\begin{aligned} x &\leq (x \to u) \rightsquigarrow u \\ &= (x \to u) \rightsquigarrow (a \to a) \\ &= a \to ((x \to u) \rightsquigarrow a). \end{aligned}$$

By Definition 2.1 (QB_4), we have $a \leq x \rightsquigarrow ((x \to u) \rightsquigarrow a)$. Since $a \in F$, it follows that $x \rightsquigarrow ((x \to u) \rightsquigarrow a) \in F$. Since X is commutative, $x \to ((x \to u) \rightsquigarrow a) \in F$. Thus, $x \in F^{pc}$. Therefore, $F^{pc} = (F^{pc})^{pc}$. □

The following example demonstrates that the condition "F is a filter of X" in Theorem 4.17 is a sufficient but not necessary for $F^{pc} = (F^{pc})^{pc}$ to hold.

Example 4.18. *Consider the structure $(X, \leq, \to, \rightsquigarrow, 0)$ from Example 3.3.*
(i) Let $F = \{-1, 0, 1\}$, we get F is not a filter of X. By the definition of F^{pc}, we have

$$\begin{aligned} F^{pc} &= \{-2, -1, 0, 1, 2\}, \\ (F^{pc})^{pc} &= \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}. \end{aligned}$$

Thus, $F^{pc} \neq (F^{pc})^{pc}$.
(ii) Let $F = \{2n + 1|n \in \mathbb{Z}\}$, we get

$$F^{pc} = \{2n|n \in \mathbb{Z}\} = (F^{pc})^{pc}.$$

But, F is not a filter of X.

Corollary 4.19. *For any normal quantum B-algebras X, the mapping $pc : \mathcal{F}(X) \to \mathcal{F}(X)$ defined by $pc(F) = F^{pc}$ is a closure operator for any $F \in \mathcal{F}(X)$.*

Proof. By Theorem 4.12, we know that $pc : \mathcal{F}(X) \to \mathcal{F}(X)$ is well-defined. It is an immediate consequence from Lemma 4.2 and Theorem 4.17. □

Lemma 4.20. *For any filter F of a normal quantum B-algebra X, the following hold:*
(1) $F_u \subseteq F^{pc}$;
(2) $(F \cup F_u)^{pc} = F^{pc}$.

Proof. (1) Let $x \to u \in F_u$ for some $x \in F$. Since F is a filter , by $x \leq (x \to u) \rightsquigarrow u$, we get $(x \to u) \rightsquigarrow u \in F$. Since X is commutative, $(x \to u) \to u \in F$. Then, it follows from $u \in F$ that $x \to u \in F^{pc}$. Hence $F_u \subseteq F^{pc}$.
(2) Since $u \in F$, according to Lemma 4.2 (2), we have $F \subseteq F^{pc}$. Moreover, from (1), we obtain $F_u \subseteq F^{pc}$. Thus, $F \subseteq F \cup F_u \subseteq F^{pc}$. Using Lemma 4.2 (1), we have $F^{pc} \subseteq (F \cup F_u)^{pc} \subseteq (F^{pc})^{pc}$. Hence, by Theorem 4.17, we get $(F \cup F_u)^{pc} = F^{pc}$. □

Example 4.21. *Consider the structure $(X, \leq, \to, \rightsquigarrow, 0)$ from Example 3.3. Let $F = \{x \in X | 0 \leq x\}$. Obviously, F is a filter of X. We get $F_0 = \{x \in X | x \leq 0\}$ and $F^{pc} = F_0^{pc} = \mathbb{R}$. Then $(F \cup F_0)^{pc} = \mathbb{R}^{pc} = \mathbb{R}$. Thus, $F_0 \subseteq F^{pc}$ and $(F \cup F_0)^{pc} = F^{pc}$.*

In the following, we examine the p-closure properties for families of closed filters in a normal quantum B-algebra X.

Theorem 4.22. *For every family $\{F_\alpha\}_{\alpha \in I}$ of closed filters of a normal quantum B-algebra X, then $(\bigcap\limits_{\alpha \in I} F_\alpha)^{pc} = \bigcap\limits_{\alpha \in I} F_\alpha^{pc}$.*

Proof. By Lemma 4.2, $(\bigcap\limits_{\alpha \in I} F_\alpha)^{pc} \subseteq F_\alpha^{pc}$ for every $\alpha \in I$. Then $(\bigcap\limits_{\alpha \in I} F_\alpha)^{pc} \subseteq \bigcap\limits_{\alpha \in I} F_\alpha^{pc}$. On the other hand, let $x \in \bigcap\limits_{\alpha \in I} F_\alpha^{pc}$. Then for every $\alpha \in I$, there exists $f_\alpha \in F_\alpha$ such that $x \to f_\alpha \in F_\alpha$ and $x \rightsquigarrow f_\alpha \in F_\alpha$. Since F_α is closed, we have $f_\alpha \to u \in F_\alpha$, and so $x \to u \in F_\alpha$ by $f_\alpha \to u \leq (x \to f_\alpha) \to (x \to u)$. Thus, $x \to u \in \bigcap\limits_{\alpha \in I} F_\alpha$. Similarly, $x \rightsquigarrow u \in \bigcap\limits_{\alpha \in I} F_\alpha$. Then we get $x \in (\bigcap\limits_{\alpha \in I} F_\alpha)^{pc}$. Therefore, $\bigcap\limits_{\alpha \in I} F_\alpha^{pc} \subseteq (\bigcap\limits_{\alpha \in I} F_\alpha)^{pc}$. Combining both inclusions, $(\bigcap\limits_{\alpha \in I} F_\alpha)^{pc} = \bigcap\limits_{\alpha \in I} F_\alpha^{pc}$. □

The commutativity of X in Theorem 4.22 is not required for this result.

Theorem 4.23. *Let $\{F_\alpha\}_{\alpha \in I}$ be a family of filters of a normal quantum B-algebra X. Then $(\bigcup\limits_{\alpha \in I} F_\alpha^{pc})^{pc} = (\bigcup\limits_{\alpha \in I} F_\alpha)^{pc}$.*

Proof. By Lemma 4.2 (2), we have $F_\alpha \subseteq F_\alpha^{pc}$ for any $\alpha \in I$, then $(\bigcup\limits_{\alpha \in I} F_\alpha) \subseteq (\bigcup\limits_{\alpha \in I} F_\alpha^{pc})$. By Lemma 4.2 (1), we have $(\bigcup\limits_{\alpha \in I} F_\alpha)^{pc} \subseteq (\bigcup\limits_{\alpha \in I} F_\alpha^{pc})^{pc}$. On the other hand, since $F_\alpha \subseteq (\bigcup\limits_{\alpha \in I} F_\alpha)$ for any $\alpha \in I$, we get $F_\alpha^{pc} \subseteq (\bigcup\limits_{\alpha \in I} F_\alpha)^{pc}$, and so, $\bigcup\limits_{\alpha \in I} F_\alpha^{pc} \subseteq$

$(\bigcup\limits_{\alpha\in I} F_\alpha)^{pc}$. Since F_α is a filter of X, using Theorem 4.17, we have $(\bigcup\limits_{\alpha\in I} F_\alpha^{pc})^{pc} \subseteq ((\bigcup\limits_{\alpha\in I} F_\alpha)^{pc})^{pc} = (\bigcup\limits_{\alpha\in I} F_\alpha)^{pc}$. Thus, $(\bigcup\limits_{\alpha\in I} F_\alpha^{pc})^{pc} = (\bigcup\limits_{\alpha\in I} F_\alpha)^{pc}$. □

Example 4.24. *Consider the structure $(X, \leq, \rightarrow, \rightsquigarrow, 0)$ from Example 3.3. Let $E = \{x \in X | 0 \leq x\}$, $F = \mathbb{R}$. Obviously, E and F are filters of X. Then we have $E^{pc} = \mathbb{R} = F^{pc}$. Thus, $(E \cap F)^{pc} = \mathbb{R} = E^{pc} \cap F^{pc}$ and $(E^{pc} \cup F^{pc})^{pc} = \mathbb{R} = (E \cup F)^{pc}$.*

Theorem 4.25. *$(\mathcal{F}_c(X), \subseteq)$ is a complete lattice where $\mathcal{F}_c(X) = \{F \subseteq X | F$ is a closed filter and $F^{pc} = F\}$.*

Proof. It is easy to prove that $(\mathcal{F}_c(X), \subseteq)$ is a poset. Let $F_\alpha \subseteq \mathcal{F}_c(X)$ for any $\alpha \in I$. By Theorem 4.22 and Theorem 4.17, we get $(\bigcap\limits_{\alpha\in I} F_\alpha)^{pc} = \bigcap\limits_{\alpha\in I} F_\alpha^{pc} = \bigcap\limits_{\alpha\in I} F_\alpha$ and $(\bigcup\limits_{\alpha\in I} F_\alpha)^{pc} = ((\bigcup\limits_{\alpha\in I} F_\alpha)^{pc})^{pc}$. Then $\bigcap\limits_{\alpha\in I} F_\alpha \subseteq \mathcal{F}_c(X)$ and $(\bigcup\limits_{\alpha\in I} F_\alpha)^{pc} \subseteq \mathcal{F}_c(X)$. By Lemma 4.2, we have $\bigcap\limits_{\alpha\in I} F_\alpha \subseteq F_\alpha \subseteq F_\alpha^{pc} \subseteq (\bigcup\limits_{\alpha\in I} F_\alpha)^{pc}$, and so, $\bigcap\limits_{\alpha\in I} F_\alpha$ is a lower bound and $(\bigcup\limits_{\alpha\in I} F_\alpha)^{pc}$ is an upper bound on F_α. Let $A \subseteq \mathcal{F}_c(X)$ such that $F_\alpha \subseteq A$ for any $\alpha \in I$, then $\bigcup\limits_{\alpha\in I} F_\alpha \subseteq A$. By $(\bigcup\limits_{\alpha\in I} F_\alpha)^{pc} \subseteq A^{pc} = A$, we get $(\bigcup\limits_{\alpha\in \mathcal{F}} F_\alpha)^{pc}$ is the supremum of F_α. Let $B \subseteq \mathcal{F}_c(X)$ such that $B \subseteq F_\alpha$ for any $\alpha \in I$, then $B \subseteq \bigcap\limits_{\alpha\in I} F_\alpha$, and so $\bigcap\limits_{\alpha\in I} F_\alpha$ is the infimum of F_α. Hence, $\mathcal{F}_c(X)$ is a complete lattice. □

5 Conclusion

In [5], it has been established that every pseudo BCI-algebra is a normal quantum B-algebra. Conversely, we characterized pseudo BCI-algebras in terms of finite normal quantum B-algebras and found an infinite normal quantum B-algebra that is not a pseudo BCI-algebra, namely Example 3.3. Subsequently, we introduced the concept of p-closure from pseudo BCI-algebras into normal quantum B-algebras, defined p-closure on non-empty subsets F of a normal quantum B-algebra X, and discussed its related properties. When F is either a subalgebra or a filter of X, we can show that F^{pc} is also a subalgebra or filter of X (Theorems 4.10 and 4.12), where the filter is not necessarily a subalgebra. It is noteworthy that in this paper, the filter structure of quantum B-algebras differs from other algebras. For instance, in pseudo BCI-algebras, the concept of filters is defined using the derivation system: $x \in F$, $x \rightarrow y \in F$ imply $y \in F$, whereas in quantum B-algebras, filters must be upper sets, meaning they must satisfy: $x \in F$, $y \in X$ such that $x \rightarrow y \in F$ imply $y \in F$. Therefore, when discussing the p-closure of filters in quantum B-algebras, we need to consider whether filters must necessarily be upper sets.

In future research, we can investigate p-closures in other logical algebras and attempt to extend other properties of pseudo BCI-algebras to infinite normal quantum B-algebras.

References

[1] M.A. Ansari, A. Haider, A.N.A. Koam, *On JU-algebras and p-closure ideals*, International Journal of Mathematics and Computer Science, 15(1):135-154, 2020.

[2] M. Botur, J. Paseka, *Filters on some classes of quantum B-algebras*, International Journal of Theoretical Physics, 54(12):4397-4409, 2015.

[3] L.C. Ciungu, *Deductive systems in unital quantum-B algebras*, Bulletin of the Belgian Mathematical Society - Simon Stevin, 29(2):173-191, 2022.

[4] L.C. Ciungu, *Quantum B-algebras with involutions*, Journal of Algebra and Its Applications, 20(12):215-233, 2021.

[5] L.C. Ciungu, *Valued quantum B-algebras*, Fuzzy Sets and Systems, 455(C):1-18, 2023.

[6] A. Dvurečenskij, M. Kuková, *Observables on quantum structures*, Information Sciences, 262:215-222, 2014.

[7] A. Dvurečenskij, S. Pulmannová, *New trends in quantum structures*, New York: Springer Science and Business Media, 2013.

[8] A. Dvurečenskij, T. Vetterlein, *Pseudo effect algebras. I. Basic properties*, International Journal of Theoretical Physics, 40(3): 685-701, 2001.

[9] A. Dvurečenskij, T. Vetterlein, *Pseudo effect algebras. II. Group Representations*, International Journal of Theoretical Physics, 40(3):703-726, 2001.

[10] G. Dymek, A. Kozanecka-Dymek, *Pseudo-BCI-logic*. Bulletin of the Section of Logic, 42(1):33-41, 2013.

[11] W.A. Dudek, Y.B. Jun, *Pseudo-BCI algebras*, East Asian Math, 24(2):187-190, 2008.

[12] P. Flondor, G. Georgescu, A Iorgulescu, *Pseudo-t-norms and pseudo-BL algebras*, Soft Computing: A Fusion of Foundations, Methodologies and Applications, 5(2):355-371, 2001.

[13] G. Georgescu, A. Iorgulescu, *Pseudo-BCK algebras: an extension of BCK-algebras*, International Conference on Discrete Mathematics and Theoretical Computer Science, 2001.

[14] G. Georgescu, A. Iorgulescu, *Pseudo MV-algebras*, Multiple-Valued Logic, 6(1-2):95-135, 2001.

[15] G. Georgescu, L. Leustean, V. Preoteasa, *Pseudo hoops*, Journal of Multiple-Valued Logic and Soft Computing, 11(1):153-184, 2005.

[16] H. Habib, *Pseudo p-closure with respect to ideals in pseudo bci-algebras*, Journal of Applied Mathematics and Informatics, 38(1-2):65-77, 2020.

[17] M. Hossein, H. Habib, *Pseudo p-closure with respect to ideals in pseudo BCI-algebras*, International Journal of Mathematics and Computer Science, 15(1):135-154, 2020.

[18] S.W. Han, R.R. Wang, X.T. Xu, *On the injective hulls of quantum B-algebras*, Fuzzy Sets and Systems, 369:114-121, 2019.

[19] S.W. Han, X.T. Xu, F. Qing, *The unitality of quantum B-algebras*, International Journal of Theoretical Physics, 57(5):1582-1590, 2018.

[20] R. Mesiar, O. Nánásiová, Z. Riečanová, et al, *Quantum structures*, Information Sciences¡ªInformatics and Computer Science, Intelligent Systems, Applications: An International Journal, 179(5):475-477, 2009.

[21] A.D. Nola, G. Georgescu, A. Iorgulescu, *Pseudo-BL algebras I,II*, Multiple-Valued Logic, 8(673):673-714, 2002.

[22] F.F. Pan, *Dual quantum B-algebras*, Soft Computing - A Fusion of Foundations, Methodologies and Applications, 23(16):6813-6817, 2019.

[23] W. Rump, *Quantum B-algebras*, Central European Journal of Mathematics, 11(11):1881-1899, 2013.

[24] W. Rump, *The completion of a quantum B-algebra*, Cah. Topol. Géom. Différ. Catég, 57(2):203-228, 2016.

[25] W. Rump, Y.C. Yang, *Hereditary arithmetics*, Journal of Algebra, 468(0):214-252, 2016.

[26] W. Rump, Y.C. Yang, *Non-commutative logical algebras and algebraic quantales*, Annals of Pure and Applied Logic, 165(2):759-785, 2014.

[27] Q.Y. Zhan, *Some operators on quantum B-algebras*, Symmetry, 13(8):1381, 2021.

[28] X.H. Zhang, R.A. Borzooei, Y.B. Jun, *Q-filters of quantum B-algebras and basic implication algebras*, Symmetry, 10(11):573-586, 2018.

[29] X.H. Zhang, Y.F. Lu, X.Y. Mao, *T-type pseudo-BCI algebras and T-type pseudo-BCI filters*, Proceedings - 2010 IEEE International Conference on Granular Computing, GrC 2010:839-844, 2010.

 Received 8 April 2025

Monadic α-filters in Monadic Residuated Lattices

Supeng Wu*
School of Mathematics and Data Science, Shaanxi University of Science and Technology, Xi'an 710021, China
wusupeng@sust.edu.cn(Corresponding author)

Abstract

The purpose of this article is to introduce and study monadic α-filters in a monadic residuated lattice. The notion of monadic co-annihilators in a monadic residuated lattice is introduced and some of their related properties are shown. Moreover, the concept of hyperarchimedean residuated lattices are proposed and some equivalent conditions are given for hyperarchimedean residuated lattices. The notion of monadic α-filters in a monadic residuated lattice is presented and some characterizations are derived for monadic α-filters. Particularly, the lattice of monadic α-filters is investigated and prime monadic α-filter theorems are established.

Keywords: Monadic residuated lattice, Monadic co-annihilator, hyperarchimedean residuated lattice, Monadic α-filter

1 Introduction

Monadic residuated lattices were proposed by Rachůnek and Šalounová [19] and further investigated by Liu et al. [15] and Wang et al. [30]. In particular, Wang et al. [30] creatively related the category of weak monadic residuated lattices and that of residuated lattices with weak universal quantifiers under the Kalman functor. Subsequently, Wang et al. [31] sorted out all of monadic algebras of t-norm based fuzzy residuated logic and then provided a new and simpler algebraic proof of completeness for monadic fuzzy predicate logic **mMTL**$\forall$ based on first order model theory and proposed the variety of monadic MTL-algebras. Monadic residuated lattices include some well-known proper classes [3, 8, 11, 18, 26, 27, 29, 31].

This work is supported by the National Natural Science Foundation of China (Grant Nos. 62473300,12401603).

The notion of α-ideals was initially introduced by Cornish [5] in distributive lattices with 0 and some important algebraic and topological representations for α-ideals were established by Cornish [6,7]. The concept of α-ideals and that of α-filters have been extended to other algebraic structures and extensively studied by some scholars [1,2,4,9,10,12,13,16,17,21,22]. Observe that the definition of α-ideals and that of α-filters are closely related to annihilators and co-annihilators, respectively. Therefore, the motivation of this article is to extend the notion of α-filters to the context of monadic residuated lattices and study some of their algebraic properties. In the present paper, we will introduce monadic co-annihilators in monadic residuated lattices and then propose the notion of monadic α-filters by means of monadic co-annihilators in monadic residuated lattices.

This paper is organized as follows: In Section 2, some definitions and facts about monadic residuated lattices are summarized. In Section 3, the notion of monadic co-annihilators in a monadic residuated lattice is presented and some relevant properties of monadic co-annihilators are investigated. Moreover, hyperarchimedean monadic residuated lattices are introduced and studied. In Section 4, the concept of monadic α-filters in a monadic residuated lattice is proposed and some algebraic characterizations of monadic α-filters are proved. In addition, a condition is shown for the lattice of all monadic α-filters to be a Boolean algebra and prime monadic α-filter theorems in monadic residuated lattices are established.

2 Preliminaries

In this section, we recall some definitions and results regarding residuated lattices and monadic residuated lattices, which will be used in the following sections.

Definition 2.1. ([15,19]) *A residuated lattice is an algebra* $(L, \wedge, \vee, \odot, \to, 0, 1)$ *of type* $(2, 2, 2, 2, 0, 0)$ *satisfying the following axioms:*

(1) $(L, \wedge, \vee, 0, 1)$ *is a bounded lattice;*

(2) $(L, \odot, 1)$ *is a commutative monoid;*

(3) $x \odot y \leq z$ *if and only if* $x \leq y \to z$*, for all* $x, y, z \in L$.

In what follows, by L we denote the universe of a residuated lattice $(L, \wedge, \vee, \odot, \to, 0, 1)$. For $x \in L$, we define: $x^* = x \to 0$, $x^0 = 1$ and $x^n = x^{n-1} \odot x$, for all $n \geq 1$.

The following four properties of a residuated lattice will be usually used in the following sections.

Proposition 2.2. ([15]) *Let L be a residuated lattice. Then the following hold for all $x, y, z \in L$:*

(1) $x \odot y = 0$ *iff* $x \leq y^*$*;*

(2) $x \odot (y \vee z) = (x \odot y) \vee (x \odot z)$*;*

(3) $x^m \vee y^n \geq (x \vee y)^{mn}$*, for some* $m, n \in \mathbb{N}$*;*

(4) $x \vee y = 1$ *implies* $x \wedge y = x \odot y$*.*

A nonempty subset F of a residuated lattice L is called a filter if the following are satisfied for all $x, y \in L$: (F_1) $x, y \in F$ implies $x \odot y \in F$; (F_2) $x \in F$ and $x \leq y$ imply $y \in F$. We denote by $\mathcal{F}(L)$ the set of all filters of a residuated lattice L.

Definition 2.3. ([19]) *A monadic residuated lattice is an algebra* $(L, \wedge, \vee, \odot, \rightarrow, \forall, \exists, 0, 1)$ *of type* $(2, 2, 2, 2, 1, 1, 0, 0)$ *satisfying the following conditions:* $(L, \wedge, \vee, \odot, \rightarrow, 0, 1)$ *is a residuated lattice and the identities hold for all $x, y \in L$:*

$(A1)$ $x \rightarrow \exists x = 1$*;*

$(A2)$ $\forall x \rightarrow x = 1$*;*

$(A3)$ $\forall(x \rightarrow \exists y) = \exists x \rightarrow \exists y$*;*

$(A4)$ $\forall(\exists x \rightarrow y) = \exists x \rightarrow \forall y$*;*

$(A5)$ $\forall(x \vee \exists y) = \forall x \vee \exists y$*;*

$(A6)$ $\exists \forall x = \forall x$*;*

$(A7)$ $\forall \forall x = \forall x$*;*

$(A8)$ $\exists(\exists x \odot \exists y) = \exists x \odot \exists y$*;*

$(A9)$ $\exists(x \odot x) = \exists x \odot \exists x$*.*

Note that it was shown by Liu [15] and Wang [30] that Castaño's and Rachůnek's axioms are equivalent for residuated lattices (not necessarily commutative).

Proposition 2.4. ([15, 19]) *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then the following properties hold for all $x, y \in L$:*

(1) $\forall 0 = 0$ *and* $\forall 1 = 1$*;*

(2) $x \leq y$ *implies* $\forall x \leq \forall y$*;*

(3) $\forall(\forall x)^* = (\forall x)^*$;

(4) $\forall(\forall x \vee \forall y) = \forall(x \vee \forall y) = \forall x \vee \forall y$;

(5) $\forall x \odot \forall y \leq \forall(x \odot y)$ *and so* $(\forall x)^n \leq \forall(x^n)$, *for all* $n \geq 1$;

(6) $\forall x = 1$ *iff* $x = 1$;

(7) $\forall(\forall x \odot \forall y) = \forall x \odot \forall y$ *and so* $\forall((\forall x)^n) = (\forall x)^n$, *for all* $n \geq 1$;

(8) $\forall(((\forall x)^n)^*) = ((\forall x)^n)^* = (\forall(\forall x)^n)^*$, *for all* $n \geq 1$;

(9) $\forall L = \{\forall x \mid x \in L\}$ *is a subalgebra of* L.

Definition 2.5. ([19]) *Let* $(L, \exists, \forall)$ *be a monadic residuated lattice. A filter* F *of* L *is called a monadic filter of* $(L, \exists, \forall)$ *if* $x \in F$ *implies* $\forall x \in F$.

Note that $x \in F$ naturally implies $\exists x \in F$ in Definition 2.5.

In a monadic residuated lattice $(L, \exists, \forall)$, we will denote by $\mathcal{F}_\forall(L)$ the set of monadic filters of $(L, \exists, \forall)$. Moreover, it is clear that $\mathcal{F}_\forall(L) \subseteq \mathcal{F}(L)$ and both $\{1\}$ and L are monadic filters. Let A be a subset of L. The least monadic filter of $(L, \exists, \forall)$ containing A is called the monadic filter generated by A and is denoted by $\langle A\rangle_\forall$. Especially, the monadic filter generated by any element $a \in L$ is called a principal monadic filter and is denoted by $\langle a\rangle_\forall$. By $F \vee G$ we denote $\langle F \cup G\rangle_\forall$, for all $F, G \in \mathcal{F}_\forall(L)$. Moreover, by $\langle F, a\rangle_\forall$ we mean $\langle F \cup \{a\}\rangle_\forall$, for $a \notin F$.

Proposition 2.6. ([15,19]) *Let* $(L, \exists, \forall)$ *be a monadic residuated lattice and* F, G *be monadic filters of* $(L, \exists, \forall)$. *Then the following hold for* $a, b \in L$:

(1) $\langle A\rangle_\forall = \{x \in L \mid x \geq \forall a_1 \odot \forall a_2 \cdots \odot \forall a_n, a_i \in A, 1 \leq i \leq n, n \geq 1\}$;

(2) $F \vee G = \langle F \cup G\rangle_\forall = \{x \in L \mid x \geq f \odot g,\ for\ some\ f \in F\ and\ g \in G\}$;

(3) $\langle F, a\rangle_\forall = \{x \in L \mid x \geq f \odot (\forall a)^n,\ for\ some\ f \in F, n \geq 1\}$;

(4) $\langle a\rangle_\forall = \langle \forall a\rangle_\forall = \{x \in L \mid x \geq (\forall a)^n, n \geq 1\}$ *and so* $\langle a\rangle_\forall = \langle a^n\rangle_\forall = \langle(\forall a)^n\rangle_\forall$ *for some* $n \geq 1$;

(5) $a \leq b$ *implies* $\langle b\rangle_\forall \subseteq \langle a\rangle_\forall$;

(6) $\langle a\rangle_\forall \cap \langle b\rangle_\forall = \langle \forall a \vee \forall b\rangle_\forall$;

(7) $\langle a\rangle_\forall \vee \langle b\rangle_\forall = \langle a \wedge b\rangle_\forall = \langle a \odot b\rangle_\forall$.

Lemma 2.7. *Let* $(L, \exists, \forall)$ *be a monadic residuated lattice and* F *be a proper monadic filter of* $(L, \exists, \forall)$. *Then the following hold:*

(1) *$a \leq b$ implies $\langle F, b\rangle_\forall \subseteq \langle F, a\rangle_\forall$;*

(2) *$\langle F, a\rangle_\forall \cap \langle F, b\rangle_\forall = \langle F, \forall a \vee \forall b\rangle_\forall$;*

(3) *$\langle F, a\rangle_\forall \vee \langle F, b\rangle_\forall = \langle F, a \odot b\rangle_\forall$;*

(4) *if $L = \langle X\rangle_\forall$, then $L = \langle A\rangle_\forall$, for some finite subset $A \subseteq X$;*

(5) *if $\{F_i \mid i \in I\}$ is a family of monadic filters of $(L, \exists, \forall)$, then $\bigvee_{i\in I} F_i = \{x \in L \mid x \geq a_{i_1} \odot \cdots \odot a_{i_n},\ for\ some\ n \geq 1, a_{i_j} \in F_{i_j}, 1 \leq j \leq n\}$.*

Proposition 2.8. ([15]) *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then $(\mathcal{F}_\forall(L), \cap, \vee, \{1\}, L)$ is a bounded distributive lattice.*

Since $\mathcal{F}(L)$ is a complete Heyting algebra, the following proposition is trivial by Lemma 2.7(5).

Proposition 2.9. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then:*

(1) *$(\mathcal{F}_\forall(L), \cap, \vee, \rightarrow_\forall, \{1\}, L)$ is a complete Heyting algebra, where $F \rightarrow_\forall G = \vee\{H \in \mathcal{F}_\forall(L) \mid F \cap H \subseteq G\}$, for all $F, G \in \mathcal{F}_\forall(L)$.*

(2) *$(\mathcal{F}_\forall(L), \cap, \vee, \{1\}, L)$ is an algebraic lattice, where the compact elements of $\mathcal{F}_\forall(L)$ are precisely the principal monadic filters of $(L,' exists, \forall)$.*

A monadic filter F of a monadic residuated lattice $(L, \exists, \forall)$ is proper if $F \neq L$. A proper monadic filter of $(L, \exists, \forall)$ is called a maximal monadic filter if it is not strictly contained in any proper monadic filter of $(L, \exists, \forall)$. The set of all maximal monadic filters of $(L, \exists, \forall)$ is denoted by $Max_\forall(L)$. The following proposition provides some characterizations of a maximal monadic filter in a monadic residuated lattice.

Proposition 2.10. ([15]) *Let $(L, \exists, \forall)$ be a monadic residuated lattice and M be a proper monadic filter of $(L, \exists, \forall)$. Then the following are equivalent:*

(1) *M is a maximal monadic filter of $(L, \exists, \forall)$;*

(2) *$\langle M, x\rangle_\forall = L$, for any $x \notin M$;*

(3) *for any $x \notin M$, there are $f \in M$ and $n \geq 1$ such that $f \odot (\forall x)^n = 0$;*

(4) *for any $x \notin M$, there is $n \geq 1$ such that $((\forall x)^n)^* \in M$.*

Definition 2.11. ([15]) *Let $(L, \exists, \forall)$ be a monadic residuated lattice. A proper monadic filter F is prime if for any monadic filter G and H, $G \cap H \subseteq F$ implies $G \subseteq F$ or $H \subseteq F$.*

By $Spec_{\forall}(L)$, we will denote the set of all prime monadic filters in a monadic residuated lattice $(L, \exists, \forall)$.

Proposition 2.12. ([15]) *Let $(L, \exists, \forall)$ be a monadic residuated lattice and P be a proper monadic filter of $(L, \exists, \forall)$. Then the following are equivalent:*

(1) *P is a prime monadic filter of $(L, \exists, \forall)$;*

(2) *for all $F, G \in \mathcal{F}_{\forall}(L)$, $F \cap G = P$ implies $F = P$ or $G = P$;*

(3) *for any $x, y \notin P$, there is $z \notin P$ and $n \geq 1$ such that $(\forall x)^n \leq z$ and $(\forall y)^n \leq z$;*

(4) *for any $x, y \notin P$, there is $z \notin P$ and $m, n \geq 1$ such that $(\forall x)^m \to z \in P$ and $(\forall y)^n \to z \in P$;*

(5) *$\forall x \vee \forall y \in P$ implies $x \in P$ or $y \in P$, for all $x, y \in L$.*

Proposition 2.13. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and P be a proper monadic filter of $(L, \exists, \forall)$. Then the following are equivalent:*

(1) *P is a prime monadic filter of $(L, \exists, \forall)$;*

(2) *for all $F \in \mathcal{F}_{\forall}(L), F \to_{\forall} P = P$ or $F \subseteq P$.*

Proof. (1) $\Rightarrow$ (2) Let P be a prime monadic filter of $(L, \exists, \forall)$. Since $\mathcal{F}_{\forall}(L)$ is a complete Heyting algebra, we have $P = (F \to_{\forall} P) \cap ((F \to_{\forall} P) \to_{\forall} P)$ for all $F \in \mathcal{F}_{\forall}(L)$, so $P = F \to_{\forall} P$ or $P = (F \to_{\forall} P) \to_{\forall} P$. If $P = (F \to_{\forall} P) \to_{\forall} P$, then $F \subseteq P$.

(2) $\Rightarrow$ (1) Let $F_1, F_2 \in \mathcal{F}_{\forall}(L)$ such that $F_1 \cap F_2 = P$, then $F_1 \subseteq F_2 \to_{\forall} P$. Suppose that $F_2 \subseteq P$. From $P = F_1 \cap F_2 \subseteq F_2$, it follows that $P = F_2$. If $F_2 \to_{\forall} P = P$, then $F_1 = P$. Thus P is a prime monadic filter. □

Condition (5) of Proposition 2.12 suggests me to focus on a special kind of subsets in a monadic residuated lattice $(L, \exists, \forall)$. A nonempty subset S of L is called $\forall$-$\vee$-closed if $x, y \in S$ implies $\forall x \vee \forall y \in S$. A monadic filter P is prime if and only if $L \setminus P$ is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$.

The subsequent lemma is an immediate consequence of Zorn's lemma.

Lemma 2.14. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. If S is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ which does not meet the monadic filter F, then S is contained in a $\forall$-$\vee$-closed subset T that is maximal with respect to the property of not meeting F.*

Theorem 2.15. ([15])(Prime monadic filter theorem) *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. If S is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ such that $F \cap S = \emptyset$, then there exists a prime monadic filter P of $(L, \exists, \forall)$ such that $F \subseteq P$ and $P \cap S = \emptyset$.*

Corollary 2.16. ([15]) *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. Then the following hold:*

(1) *If $a \notin F$, then there is $P \in Spec_\forall(L)$ such that $F \subseteq P$ and $a \notin P$;*

(2) *if $a < 1$, then there is $P \in Spec_\forall(L)$ such that $a \notin P$.*

Proposition 2.17. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. Then the following hold:*

(1) *If $X \nsubseteq F$, then there is $P \in Spec_\forall(L)$ such that $F \subseteq P$ and $X \nsubseteq P$;*

(2) *any proper monadic filter F is always the intersection of all prime monadic filter containing F, that is, $F = \bigcap\{P \in Spec_\forall(L) \mid F \subseteq P\}$;*

(3) $\bigcap Spec_\forall(L) = \{1\}$.

Proof. (1) If $X \nsubseteq F$, then there is $a \in X$ and $a \notin F$. From Corollary 2.16(1) it follows that there is a prime monadic filter P of $(L, \exists, \forall)$ such that $F \subseteq P$ and $a \notin P$. Thus $F \subseteq P$ and $X \nsubseteq P$.

(2) Obviously, $F \subseteq \bigcap\{P \in Spec_\forall(L) \mid F \subseteq P\}$. Conversely, suppose that $x \notin F$. By Corollary 2.16(1), there is a prime monadic filter P of $(L, \exists, \forall)$ such that $F \subseteq P$ and $x \notin P$. So $x \notin \bigcap\{P \in Spec_\forall(L) \mid F \subseteq P\}$. This means that $\bigcap\{P \in Spec_\forall(L) \mid F \subseteq P\} \subseteq F$. Thus, $F = \bigcap\{P \in Spec_\forall(L) \mid F \subseteq P\}$.

(3) Take $F = \{1\}$. By (2), we deduce that $\bigcap Spec_\forall(L) = \{1\}$. □

3 Monadic co-annihilators

In this section, the concept of monadic co-annihilators is presented in a monadic residuated lattice, which can be regarded as a generalization of the notion of co-annihilators in a residuated lattice [20] and that of monadic co-annihilators in a monadic BL-algebra [28]. Some basic and useful properties concerning monadic co-annihilators are obtained.

Definition 3.1. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and A be a subset of L. The set $A^{\perp}_{\forall} = \{x \in L \mid \forall x \vee \forall a = 1, for\ all\ a \in A\}$ is called the monadic co-annihilator of A.*

Notice that $A^{\perp}_{\forall} = \{x \in L \mid x \vee \forall a = 1, for\ all\ a \in A\} = \{x \in L \mid \forall x \vee a = 1, for\ all\ a \in A\}$ by Proposition 2.4(4) and (6). Particularly, $\{x\}^{\perp}_{\forall}$ is simply denoted by $x^{\perp}_{\forall}$. It is easily proved that $0^{\perp}_{\forall} = L^{\perp}_{\forall} = \{1\}$ and $1^{\perp}_{\forall} = \emptyset^{\perp}_{\forall} = L$.

Example 3.2. *Consider the residuated lattice $L = \{0, a, b, c, 1\}$ and $0 < a < b, c < 1$ from [15]. The operations $\odot$ and $\to$ are defined on L as follows: $x \odot y = x \wedge y$, $x \to y = max\{z \in L \mid x \odot z \leq y\}$, for all $x, y \in L$. The operations $\forall, \exists$ are defined on L as follows:*

x	0	a	b	c	1
$\forall_1 x$	0	a	a	a	1
$\forall_2 x$	0	0	0	0	1
$\forall_3 x$	0	a	b	c	1

x	0	a	b	c	1
$\exists_1 x$	0	a	1	1	1
$\exists_2 x$	0	1	1	1	1
$\exists_3 x$	0	a	b	c	1

One can check that $(L, \forall_1, \exists_1)$, $(L, \forall_2, \exists_2)$ and $(L, \forall_3, \exists_3)$ are monadic residuated lattices.

Clearly, $A^{\perp}_{\forall_1} = \{1\}$ and $A^{\perp}_{\forall_2} = \{1\}$, for any nonempty subset $A \neq \{1\}$ of L. Routine calculation shows that $\{b, 1\}^{\perp}_{\forall_3} = \{c, 1\}, \{c, 1\}^{\perp}_{\forall_3} = \{b, 1\}$ and $\{a, 1\}^{\perp}_{\forall_3} = \{a, c, 1\}^{\perp}_{\forall_3} = \{a, b, 1\}^{\perp}_{\forall_3} = \{b, c, 1\}^{\perp}_{\forall_3} = \{a, b, c, 1\}^{\perp}_{\forall_3} = L^{\perp}_{\forall_3} = \{1\}$.

Proposition 3.3. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then the following properties hold for $A, B \subseteq L$:*

(1) *$A^{\perp}_{\forall}$ is a monadic filter of $(L, \exists, \forall)$;*

(2) *$A \subseteq B$ implies $B^{\perp}_{\forall} \subseteq A^{\perp}_{\forall}$ and $(A^{\perp}_{\forall})^{\perp}_{\forall} \subseteq (B^{\perp}_{\forall})^{\perp}_{\forall}$;*

(3) *$A \subseteq (A^{\perp}_{\forall})^{\perp}_{\forall}$;*

(4) *$A^{\perp}_{\forall} = ((A^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall}$;*

(5) *$A^{\perp}_{\forall} \subseteq B^{\perp}_{\forall}$ if and only if $(B^{\perp}_{\forall})^{\perp}_{\forall} \subseteq (A^{\perp}_{\forall})^{\perp}_{\forall}$;*

(6) *$A^{\perp}_{\forall} = (\langle A \rangle_{\forall})^{\perp}_{\forall}$;*

(7) *$\langle A \rangle_{\forall} \cap A^{\perp}_{\forall} = \{1\}$ and $A^{\perp}_{\forall} \cap (A^{\perp}_{\forall})^{\perp}_{\forall} = \{1\}$;*

(8) *$((A \cap B)^{\perp}_{\forall})^{\perp}_{\forall} = (A^{\perp}_{\forall})^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall}$;*

(9) *$\bigcap_{i \in I} (A_i)^{\perp}_{\forall} = (\bigcup_{i \in I} A_i)^{\perp}_{\forall}$, for $\{A_i \mid i \in I\} \subseteq L$;*

(10) *$A^{\perp}_{\forall} = L$ iff $A = \{1\}$.*

Proof. (1) It is easily proved that $A^{\perp}_{\forall}$ is a filter of L. If $x \in A^{\perp}_{\forall}$, then $\forall x \vee \forall a = 1$ for all $a \in A$. By Definition 2.3(A7), $\forall\forall x \vee \forall a = 1$ for all $a \in A$, which means that $\forall x \in A^{\perp}_{\forall}$. We deduce that $A^{\perp}_{\forall}$ is a monadic filter of L.

(2) Let $A \subseteq B$ and $x \in B^{\perp}_{\forall}$. Then $\forall x \vee \forall b = 1$ for all $b \in B$. Hence $\forall x \vee \forall a = 1$ for all $a \in A$, which implies that $x \in A^{\perp}_{\forall}$. Thus $B^{\perp}_{\forall} \subseteq A^{\perp}_{\forall}$ and so $(A^{\perp}_{\forall})^{\perp}_{\forall} \subseteq (B^{\perp}_{\forall})^{\perp}_{\forall}$.

(3) Let $a \in A$. For all $x \in A^{\perp}_{\forall}$, we have $\forall x \vee \forall a = 1$, so $a \in (A^{\perp}_{\forall})^{\perp}_{\forall}$. Hence, $A \subseteq (A^{\perp}_{\forall})^{\perp}_{\forall}$.

(4) By (3), we have $A^{\perp}_{\forall} \subseteq ((A^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall}$. Conversely, by (3) and (2), we have $((A^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} \subseteq A^{\perp}_{\forall}$. Thus, $A^{\perp}_{\forall} = ((A^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall}$.

(5) This follows from (2) and (4).

(6) Since $A \subseteq \langle A \rangle_{\forall}$, by (2) it follows that $(\langle A \rangle_{\forall})^{\perp}_{\forall} \subseteq A^{\perp}_{\forall}$. If $x \in A^{\perp}_{\forall}$, then $\forall x \vee \forall a = 1$ for all $a \in A$. For all $y \in \langle A \rangle_{\forall}$, there exist $a_1, a_2 \cdots, a_n \in A$ such that $y \geq \forall a_1 \odot \cdots \odot \forall a_n$. So $\forall y \geq \forall(\forall a_1 \odot \cdots \odot \forall a_n)$. From Proposition 2.2(3) and Proposition 2.4(7) it follows that $\forall x \vee \forall y \geq \forall x \vee \forall(\forall a_1 \odot \cdots \odot \forall a_n) = \forall x \vee (\forall a_1 \odot \cdots \odot \forall a_n) \geq (\forall x \vee \forall a_1) \odot \cdots \odot (\forall x \vee \forall a_n) = 1$ and so $x \in (\langle A \rangle_{\forall})^{\perp}_{\forall}$. Hence $A^{\perp}_{\forall} \subseteq (\langle A \rangle_{\forall})^{\perp}_{\forall}$. We conclude that $A^{\perp}_{\forall} = (\langle A \rangle_{\forall})^{\perp}_{\forall}$.

(7) For all $x \in \langle A \rangle_{\forall} \cap A^{\perp}_{\forall}$, we have $x \in \langle A \rangle_{\forall}$ and $x \in A^{\perp}_{\forall}$. Thus $\forall x = \forall x \vee \forall x = 1$, which means that $x = 1$.

(8) Since $A \cap B \subseteq A, B$, by (2) we have $((A \cap B)^{\perp}_{\forall})^{\perp}_{\forall} \subseteq (A^{\perp}_{\forall})^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall}$. Conversely, from (7) it follows that $(A \cap B) \cap (A \cap B)^{\perp}_{\forall} = \{1\}$, so $A \cap (A \cap B)^{\perp}_{\forall} \subseteq B^{\perp}_{\forall}$. Applying (7) again, we have $A \cap (A \cap B)^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall} \subseteq B^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall} = \{1\}$, so $(A \cap B)^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall} \subseteq A^{\perp}_{\forall}$, which implies that $(A \cap B)^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall} \cap (A^{\perp}_{\forall})^{\perp}_{\forall} \subseteq A^{\perp}_{\forall} \cap (A^{\perp}_{\forall})^{\perp}_{\forall} = \{1\}$. Hence $(A^{\perp}_{\forall})^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall} \subseteq ((A \cap B)^{\perp}_{\forall})^{\perp}_{\forall}$. We conclude that $((A \cap B)^{\perp}_{\forall})^{\perp}_{\forall} = (A^{\perp}_{\forall})^{\perp}_{\forall} \cap (B^{\perp}_{\forall})^{\perp}_{\forall}$.

(9) It follows from (2) that $(\bigcup_{i \in I} A_i)^{\perp}_{\forall} \subseteq (A_i)^{\perp}_{\forall}$ for all $i \in I$, so $(\bigcup_{i \in I} A_i)^{\perp}_{\forall} \subseteq \bigcap_{i \in I}(A_i)^{\perp}_{\forall}$. Conversely, if $x \in \bigcap_{i \in I}(A_i)^{\perp}_{\forall}$, then $x \in (A_i)^{\perp}_{\forall}$ for all $i \in I$. Hence, $\forall x \vee \forall a = 1$ for all $a \in A_i$ and $i \in I$, which implies that $x \in (\bigcup_{i \in I} A_i)^{\perp}_{\forall}$. Thus, $\bigcap_{i \in I}(A_i)^{\perp}_{\forall} = (\bigcup_{i \in I} A_i)^{\perp}_{\forall}$.

(10) Let $A^{\perp}_{\forall} = L$. By (3), we have $A \subseteq (A^{\perp}_{\forall})^{\perp}_{\forall} = L^{\perp}_{\forall} = \{1\}$, so $A = \{1\}$. Conversely, suppose that $A = \{1\}$. Then $A^{\perp}_{\forall} = (\{1\})^{\perp}_{\forall} = L$. □

The set of all monadic co-annihilators of a monadic residuated lattice $(L, \exists, \forall)$ is denoted by $\mathcal{CA}_{\forall}(L)$ and $\mathcal{CA}_{\forall}(L) = \{A^{\perp}_{\forall} \mid A \subseteq L\}$. From Proposition 3.3 it follows that $\mathcal{CA}_{\forall}(L) = \{\langle A \rangle_{\forall} \in \mathcal{F}_{\forall}(L) \mid \langle A \rangle_{\forall} = (A^{\perp}_{\forall})^{\perp}_{\forall}\}$. We define a join operation $\sqcup$ on $\mathcal{CA}_{\forall}(L)$: $A^{\perp}_{\forall} \sqcup B^{\perp}_{\forall} = (A^{\perp}_{\forall} \cap B^{\perp}_{\forall})^{\perp}_{\forall}$ for $A, B \subseteq L$. One can check that $(\mathcal{CA}_{\forall}(L), \cap, \sqcup, \{1\}, L)$ is a Boolean algebra.

The subsequent lemma is a direct consequence of Proposition 2.6 and Proposition 3.3.

Lemma 3.4. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then the following hold for all $a, b \in L$:*

(1) $a \le b$ *implies* $a^{\perp}_{\forall} \subseteq b^{\perp}_{\forall}$ *and* $(b^{\perp}_{\forall})^{\perp}_{\forall} \subseteq (a^{\perp}_{\forall})^{\perp}_{\forall}$*;*

(2) $(a^{\perp}_{\forall})^{\perp}_{\forall} = \{x \in L \mid \forall x \vee \forall y = 1$ *for any* $y \in L$ *such that* $\forall y \vee \forall a = 1\}$*;*

(3) $a \in (a^{\perp}_{\forall})^{\perp}_{\forall}$ *and* $a^{\perp}_{\forall} = ((a^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall}$*;*

(4) $b \in a^{\perp}_{\forall}$ *implies* $(a^{\perp}_{\forall})^{\perp}_{\forall} \subseteq b^{\perp}_{\forall}$ *and* $(b^{\perp}_{\forall})^{\perp}_{\forall} \subseteq a^{\perp}_{\forall}$*;*

(5) $a^{\perp}_{\forall} = (\langle a \rangle_{\forall})^{\perp}_{\forall} = (\forall a)^{\perp}_{\forall}$*;*

(6) $a^{\perp}_{\forall} \cap b^{\perp}_{\forall} = (a \odot b)^{\perp}_{\forall} = (a \wedge b)^{\perp}_{\forall}$*;*

(7) $a^{\perp}_{\forall} = (a^2)^{\perp}_{\forall} = (a^n)^{\perp}_{\forall}$*, for some* $n \in \mathbb{N}, n \ge 2$*;*

(8) $(a^{\perp}_{\forall})^{\perp}_{\forall} \cap (b^{\perp}_{\forall})^{\perp}_{\forall} = ((\forall a \vee \forall b)^{\perp}_{\forall})^{\perp}_{\forall}$*;*

(9) $a^{\perp}_{\forall} \sqcup b^{\perp}_{\forall} = (\forall a \vee \forall b)^{\perp}_{\forall}$*;*

(10) $(a^{\perp}_{\forall})^{\perp}_{\forall} \sqcup (b^{\perp}_{\forall})^{\perp}_{\forall} = ((a \odot b)^{\perp}_{\forall})^{\perp}_{\forall} = ((a \wedge b)^{\perp}_{\forall})^{\perp}_{\forall}$*;*

(11) $\langle a \rangle_{\forall} \cap a^{\perp}_{\forall} = \{1\}$ *and* $a^{\perp}_{\forall} \cap (a^{\perp}_{\forall})^{\perp}_{\forall} = \{1\}$.

(12) $a^{\perp}_{\forall} = L$ *if and only if* $a = 1$.

Proof. (1) If $a \le b$, then by Proposition 2.6(5) it follows that $\langle b \rangle_{\forall} \subseteq \langle a \rangle_{\forall}$. From Proposition 3.3(2) we have $a^{\perp}_{\forall} \subseteq b^{\perp}_{\forall}$ and so $(b^{\perp}_{\forall})^{\perp}_{\forall} \subseteq (a^{\perp}_{\forall})^{\perp}_{\forall}$.

(2) It is clear.

(3) This follows from Proposition 3.3(3) and (4).

(4) This follows from Proposition 3.3(2).

(5) By Proposition 2.6(4) and Proposition 3.3(6), $a^{\perp}_{\forall} = (\langle a \rangle_{\forall})^{\perp}_{\forall} = (\forall a)^{\perp}_{\forall}$.

(6) Since $a \odot b \le a \wedge b \le a, b$, by (1) we have $(a \odot b)^{\perp}_{\forall} \subseteq (a \wedge b)^{\perp}_{\forall} \subseteq a^{\perp}_{\forall} \cap b^{\perp}_{\forall}$. Conversely, if $x \in a^{\perp}_{\forall} \cap b^{\perp}_{\forall}$, then $x \in a^{\perp}_{\forall}$ and $x \in b^{\perp}_{\forall}$. This means that $\forall x \vee \forall a = 1$ and $\forall x \vee \forall b = 1$. By Proposition 2.4(5) and Proposition 2.2(3), we have $\forall x \vee \forall (a \odot b) \ge \forall x \vee (\forall a \odot \forall b) \ge (\forall x \vee \forall a) \odot (\forall x \vee \forall b) = 1$. It follows that $x \in (a \odot b)^{\perp}_{\forall}$. Thus $a^{\perp}_{\forall} \cap b^{\perp}_{\forall} = (a \odot b)^{\perp}_{\forall} = (a \wedge b)^{\perp}_{\forall}$.

(7) By (6), we have $a^{\perp}_{\forall} = (a^2)^{\perp}_{\forall} = (a^n)^{\perp}_{\forall}$, for some $n \ge 2$.

(8) By Proposition 3.3((6) and (8)) and Proposition 2.6(6), we have $(a^{\perp}_{\forall})^{\perp}_{\forall} \cap (b^{\perp}_{\forall})^{\perp}_{\forall} = ((\langle a \rangle_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} \cap ((\langle b \rangle_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} = ((\langle a \rangle_{\forall} \cap \langle b \rangle_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} = ((\langle \forall a \vee \forall b \rangle_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} = ((\forall a \vee \forall b)^{\perp}_{\forall})^{\perp}_{\forall}$.

(9) Applying (8) and (3), we have $a^{\perp}_{\forall} \sqcup b^{\perp}_{\forall} = ((a^{\perp}_{\forall})^{\perp}_{\forall} \cap (b^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} = (((\forall a \vee \forall b)^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} = (\forall a \vee \forall b)^{\perp}_{\forall}$.

(10) By (3) and (6), $(a^{\perp}_{\forall})^{\perp}_{\forall} \sqcup (b^{\perp}_{\forall})^{\perp}_{\forall} = (((a^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} \cap ((b^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall})^{\perp}_{\forall} = (a^{\perp}_{\forall} \cap b^{\perp}_{\forall})^{\perp}_{\forall} = ((a \odot b)^{\perp}_{\forall})^{\perp}_{\forall} = ((a \wedge b)^{\perp}_{\forall})^{\perp}_{\forall}$.

(11) It follows from Proposition 3.3(7).

(12) If $a = 1$, then $a_{\forall}^{\perp} = L$. Conversely, if $a_{\forall}^{\perp} = L$, then $0 \in a_{\forall}^{\perp}$. Thus $\forall a = \forall 0 \vee \forall a = 1$ and so $a = 1$. $\Box$

For any element $x \in L$, $x_{\forall}^{\perp}$ is called a monadic co-annulet of $(L, \exists, \forall)$. The set of all monadic co-annulets of $(L, \exists, \forall)$ is denoted by $\mathcal{L}_{\forall}^{\perp}$ and $\mathcal{L}_{\forall}^{\perp} = \{x_{\forall}^{\perp} \mid x \in L\}$. From Lemma 3.4(6) and (9) it follows that $(\mathcal{L}_{\forall}^{\perp}, \cap, \sqcup, \{1\}, L)$ is a sublattice of $(\mathcal{CA}_{\forall}(L), \cap, \sqcup, \{1\}, L)$.

Note that in Example 3.2 one can easily check that $\mathcal{L}_{\forall_1}^{\perp} = \mathcal{L}_{\forall_2}^{\perp} = \{\{1\}, L\}$ and $\mathcal{L}_{\forall_3}^{\perp} = \{\{1\}, \{b, 1\}, \{c, 1\}, L\}$.

Two characterizations of monadic co-annihilators will be provided by means of prime monadic filters in a monadic residuated lattice.

Proposition 3.5. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and $A \subseteq L$. Then the following hold:*

(1) $A_{\forall}^{\perp} = \bigcap\{P \in Spec_{\forall}(L) \mid A_{\forall}^{\perp} \subseteq P\}$;

(2) $A_{\forall}^{\perp} = \bigcap\{P \in Spec_{\forall}(L) \mid A \nsubseteq P\}$.

Proof. (1) Since $A_{\forall}^{\perp} \in \mathcal{F}_{\forall}(L)$, by Proposition 2.17(2) we have $A_{\forall}^{\perp} = \bigcap\{P \in Spec_{\forall}(L) \mid A_{\forall}^{\perp} \subseteq P\}$.

(2) Let $x \in A_{\forall}^{\perp}$ and $P \in Spec_{\forall}(L)$ such that $A \nsubseteq P$. If $a \in A \setminus P$, then $\forall x \vee \forall a = 1$. Since $1 \in P \in Spec_{\forall}(L)$, we have $x \in P$. This implies that $x \in \bigcap\{P \in Spec_{\forall}(L) \mid A \nsubseteq P\}$. Conversely, suppose that $x \in \bigcap\{P \in Spec_{\forall}(L) \mid A \nsubseteq P\}$. Then $x \in P$ for all $P \in Spec_{\forall}(L)$ satisfying $A \nsubseteq P$, which implies that $\forall x \in P$. Since P is an upset, we have $\forall x \vee \forall a \in P$, for all $a \in A$. On the other hand, for all $a \in A$, we have $\forall a \in \langle A \rangle_{\forall}$ and so $\forall a \vee \forall x \in \langle A \rangle_{\forall}$. From Proposition 2.17(2) it follows that $\forall x \vee \forall a \in \langle A \rangle_{\forall} = \bigcap\{P \in Spec_{\forall}(L) \mid A \subseteq P\}$. Therefore, $\forall x \vee \forall a \in P$ for all $P \in Spec_{\forall}(L)$. By Proposition 2.17(3), we have $\forall x \vee \forall a \in \bigcap Spec_{\forall}(L) = \{1\}$. This means that $x \in A_{\forall}^{\perp}$. $\Box$

Definition 3.6. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. A prime monadic filter P of $(L, \exists, \forall)$ is called a minimal prime monadic filter belonging to F or an F-minimal prime monadic filter if for any prime monadic filter Q of $(L, \exists, \forall)$, $F \subseteq Q \subseteq P$ implies $Q = P$. In particular, a minimal prime monadic filter belonging to $\{1\}$ is simply called a minimal prime monadic filter, that is, if for any prime monadic filter Q of $(L, \exists, \forall)$, $Q \subseteq P$ implies $Q = P$.*

The set of all minimal prime monadic filters of $(L, \exists, \forall)$ is denoted by $Min_{\forall}(L)$.

Example 3.7. (1) *Consider the monadic residuated lattice $(L, \exists_1, \forall_1)$ in Example 3.2. It is easily checked that both $\{1\}$ and $\{a, b, c, 1\}$ are prime monadic filters of*

$(L, \exists_1, \forall_1)$. *Moreover,* $\{1\}$ *is a minimal prime monadic filter and* $\{a, b, c, 1\}$ *is a maximal monadic filter.*

(2) *Consider the monadic residuated lattice* $L = \{0, a, b, c, d, 1\}$ *with* $0 < a < b < c, d < 1$ *from [15]. The operations* $\odot, \rightarrow$ *on* L *are given as follows: For all* $x, y \in L$, $x \odot y = x \wedge y$, $x \rightarrow y = max\{z \in L \mid x \odot z \leq y\}$. *The operations* $\forall, \exists$ *are defined on* L *as follows:*

$$\forall 0 = \forall a = 0, \forall b = b, \forall c = c, \forall d = d, \forall 1 = 1;$$

$$\exists 0 = 0, \exists a = \exists b = b, \exists c = c, \exists d = d, \exists 1 = 1.$$

Obverse that $F_0 = \{1\}$ *is not a prime monadic filter. Set* $F_1 = \{c, 1\}$, $F_2 = \{d, 1\}$ *and* $F_3 = \{b, c, d, 1\}$. *One can easily check that* F_1, F_2 *and* F_3 *are all prime monadic filters. Moreover,* F_1 *and* F_2 *are* $\{1\}$*-minimal prime monadic filters and* F_3 *is a maximal monadic filter.*

The next theorem is analogous to the one given for minimal prime ideals in commutative semigroups by Kist [14].

Theorem 3.8. *(Minimal prime monadic filter theorem) Let* $(L, \exists, \forall)$ *be a monadic residuated lattice and* F *be a monadic filter of* $(L, \exists, \forall)$. *A subset* P *of* L *is an* F*-minimal prime monadic filter if and only if* $L \setminus P$ *is a* $\forall$-$\vee$*-closed subset of* $(L, \exists, \forall)$ *which is maximal with respect to the property of not meeting* F.

Proof. Suppose that P is a subset of L such that $L \setminus P$ is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ which is maximal with respect to the property of not meeting F. From Theorem 2.15 it follows that there is a prime monadic filter Q such that $F \subseteq Q$ and $(L \setminus P) \cap Q = \emptyset$, so $L \setminus P \subseteq L \setminus Q$. By the maximality of $L \setminus P$ and $(L \setminus Q) \cap F = \emptyset$, we have $L \setminus P = L \setminus Q$, so $P = Q$. Thus P is an F-minimal prime monadic filter.

Conversely, let P be an F-minimal prime monadic filter of $(L, \exists, \forall)$. Then $L \setminus P$ is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ and $(L \setminus P) \cap F = \emptyset$. According to Lemma 2.14, there is a $\forall$-$\vee$-closed subset S of $(L, \exists, \forall)$ which is maximal with respect to the property of not meeting F and $L \setminus P \subseteq S$. Hence $L \setminus S$ is an F-minimal prime monadic filter and $(L \setminus P) \cap (L \setminus S) = \emptyset$, which implies that $L \setminus S \subseteq P$. By the minimality of P, we have $L \setminus S = P$, so $L \setminus P = S$, which means that $L \setminus P$ is a $\forall$-$\vee$-closed subset which is maximal with respect to the property of not meeting F. □

Lemma 3.9. *Let* $(L, \exists, \forall)$ *be a monadic residuated lattice and* F *be a monadic filter of* $(L, \exists, \forall)$. *If* P *is a prime monadic filter containing* F, *then there is an* F*-minimal prime monadic filter contained in* P.

Proof. Let P be a prime monadic filter containing F. Then $L \setminus P$ is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ such that $(L \setminus P) \cap F = \emptyset$. By Lemma 2.14, there is a $\forall$-$\vee$-closed subset S of $(L,\exists,\forall)$ which is maximal with respect to the property of not meeting F and $L\setminus P\subseteq S$. From Theorem 3.8 it follows that $L\setminus S$ is an F-minimal prime monadic filter contained in P. $\square$

The following corollary is an immediate consequence of Corollary 2.16, Proposition 2.17 and Lemma 3.9.

Corollary 3.10. *Let $(L,\exists,\forall)$ be a monadic residuated lattice and F be a monadic filter of $(L,\exists,\forall)$. Then the following hold for $A,X\subseteq L$:*

(1) *If $X\nsubseteq F$, then there is an F-minimal prime monadic filter P of $(L,\exists,\forall)$ such that $F\subseteq P$ and $X\nsubseteq P$;*

(2) *if $a\notin F$, then there is an F-minimal prime monadic filter P of $(L,\exists,\forall)$ such that $F\subseteq P$ and $a\notin P$;*

(3) *if $a<1$, then there is a minimal prime monadic filter P of $(L,\exists,\forall)$ such that $a\notin P$;*

(4) *any proper monadic filter F is always the intersection of all F-minimal prime monadic filter containing F, that is, $F=\bigcap\{P\in Min_{\forall}(L)\mid F\subseteq P\}$;*

(5) $\bigcap Min_{\forall}(L)=\{1\}$;

(6) $A^{\perp}_{\forall}=\bigcap\{P\in Min_{\forall}(L)\mid A\nsubseteq P\}$.

Theorem 3.11. *Let $(L,\exists,\forall)$ be a monadic residuated lattice and F be a monadic filter of $(L,\exists,\forall)$. If P is a prime monadic filter of $(L,\exists,\forall)$ such that $F\subseteq P$, then the following are equivalent:*

(1) *P is an F-minimal prime monadic filter;*

(2) *for any $a\in P$, there is $b\notin P$ such that $\forall a\vee\forall b\in F$.*

Proof. (1) $\Rightarrow$ (2) Let $a\in P$ and construct the set $S_a=\{x\in L\mid\forall x\leq\forall a\vee\forall b,\ \textit{for some}\ b\notin P\}$. Since $\forall a=\forall a\vee\forall 0$, we have $a\in S_a$. Since $\forall b\leq\forall a\vee\forall b$ for any $b\in L\setminus P$, we have $b\in S_a$, which means that $L\setminus P\subseteq S_a$. Let $x,y\in S_a$. Then there are $b,c\notin P$ such that $\forall x\leq\forall a\vee\forall b$ and $\forall y\leq\forall a\vee\forall c$. So $\forall x\vee\forall y\leq\forall a\vee\forall b\vee\forall a\vee\forall c=\forall a\vee(\forall b\vee\forall c)$. Applying Proposition 2.4(4), we have $\forall(\forall x\vee\forall y)\leq\forall a\vee\forall(\forall b\vee\forall c)$. Since P is prime and $b,c\notin P$, it follows that $\forall b\vee\forall c\notin P$, so $\forall x\vee\forall y\in S_a$. This means that S_a is a $\forall$-$\vee$-closed subset.

Next we shall prove that $F\cap S_a\neq\emptyset$. Assume that $F\cap S_a=\emptyset$. By Theorem 2.15 there is a prime monadic filter R of $(L,\exists,\forall)$ such that $F\subseteq R$ and $R\cap S_a=\emptyset$. From $a\in S_a$ and $R\cap S_a=\emptyset$, we have $a\notin R$, so $R\neq P$ as $a\in P$. We now prove that

$R \subseteq P$. If $R \nsubseteq P$, then there is $x \in R$ such that $x \notin P$. Since R is a monadic filter, we have $\forall a \vee \forall x \in R$. From $x \notin P$ and $L \setminus P \subseteq S_a$, we have $x \in S_a$, so there exists $b \in L \setminus P$ such that $\forall a \vee \forall b \geq \forall x$, which means that $\forall a \vee \forall b \geq \forall a \vee \forall x = \forall(\forall a \vee \forall x)$ by Proposition 2.4(4). Thus $\forall a \vee \forall x \in S_a$ and so $\forall a \vee \forall x \in R \cap S_a \neq \emptyset$, a contradiction. Hence $R \subsetneq P$, which contradicts the fact that P is a minimal prime monadic filter such that $F \subseteq P$. Therefore, $F \cap S_a \neq \emptyset$, so there is some $u \in L$ such that $u \in F$ and $u \in S_a$. From $u \in S_a$, it follows that there is $b \notin P$ such that $\forall a \vee \forall b \geq \forall u$. Since F is a monadic filter and $u \in F$, we have $\forall u \in F$. So $\forall a \vee \forall b \in F$ as F is an upset.

(2) $\Rightarrow$ (1) Suppose that $Q \in Spec_{\forall}(L)$ such that $F \subseteq Q \subseteq P$. We prove that $P \subseteq Q$. Consider $x \in P$. By the hypothesis there is $y \notin P$ such that $\forall x \vee \forall y \in F \subseteq Q$. From $y \notin P$ and $Q \subseteq P$, we have $y \notin Q$, which implies that $x \in Q$. Hence $P \subseteq Q$, so $Q = P$. This shows that P is a minimal prime monadic filter such that $F \subseteq P$. □

Let F be a monadic filter of a monadic residuated lattice $(L, \exists, \forall)$. Suppose that P is a prime monadic filter containing F. We define the set $D^F_{\forall}(P) = \{x \in L \mid \forall x \vee \forall y \in F, \ for\ some\ y \notin P\}$ and it is evident that $D^F_{\forall}(P) = \{x \in L \mid x^{\perp_F}_{\forall} \nsubseteq P\} = \bigcup_{y \notin P} y^{\perp_F}_{\forall}$, where $x^{\perp_F}_{\forall} = \{a \in L \mid \forall a \vee \forall x \in F\}$, which represents the monadic co-annihilator of $\{x\}$ with respect to F. One can easily verify that $D^F_{\forall}(P) \subseteq P$. We now restate Theorem 3.11 by means of the set $D^F_{\forall}(P)$ in the subsequent theorem.

The next theorem is similar to the one obtained for minimal prime ideals in a distributive lattice with 0 by Speed [25] and the one derived for minimal prime filters in a residuated lattice by Rasouli [21, 22].

Theorem 3.12. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. If P is a prime monadic filter of $(L, \exists, \forall)$ such that $F \subseteq P$, then the following are equivalent:*

(1) *P is an F-minimal prime monadic filter;*

(2) *$D^F_{\forall}(P) = P$;*

(3) *for all $x \in L$, P contains exactly one of x or $x^{\perp_F}_{\forall}$.*

Proof. It is an immediate consequence of Theorem 3.11. □

Corollary 3.13. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and P be a minimal prime monadic filter of $(L, \exists, \forall)$. Then for all $a \in P$ there is $b \notin P$ such that $\forall a \vee \forall b = 1$.*

Proof. Take $F = \{1\}$. It is a direct consequence of Theorem 3.11. □

Let $(L, \exists, \forall)$ be a monadic residuated lattice and P be a prime monadic filter of $(L, \exists, \forall)$. $D_{\forall}^{\{1\}}(P)$ is simply denoted by $D_{\forall}(P)$ and $D_{\forall}(P) \subseteq P$.

Corollary 3.14. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and P be a prime monadic filter of $(L, \exists, \forall)$. Then the following are equivalent:*

(1) *P is a minimal prime monadic filter;*

(2) *$D_{\forall}(P) = P$;*

(3) *for all $x \in L$, P contains exactly one of x or $x_{\forall}^{\perp}$.*

Proof. This follows from Theorem 3.12. □

In order to investigate further properties of the lattice $\mathcal{F}_{\forall}(L)$, we focus on a special kind of elements in a monadic residuated lattice $(L, \exists, \forall)$. An element $a \in L$ is called $\forall$-complemented if there exists $b \in L$ such that $\forall a \vee \forall b = 1$ and $\forall a \wedge \forall b = 0$, which corresponds to the complement of an element in a residuated lattice. The set of all $\forall$-complemented elements of $(L, \exists, \forall)$ is called the monadic boolean center and is denoted by $\mathcal{B}_{\forall}(L)$.

Notice that both 0 and 1 are always $\forall$-complemented in any monadic residuated lattice $(L, \exists, \forall)$.

Let $(L, \exists, \forall)$ be a monadic residuated lattice. The set of all principle monadic filters of $(L, \exists, \forall)$ is denoted by $\mathcal{P}_{\forall}(L)$ and the set of all complemented elements of the bounded distributive lattice $\mathcal{F}_{\forall}(L)$ is denoted by $\mathcal{B}(\mathcal{F}_{\forall}(L))$.

Lemma 3.15. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then the following are equivalent:*

(1) *$F \in \mathcal{B}(\mathcal{F}_{\forall}(L))$;*

(2) *$F \vee F_{\forall}^{\perp} = L$;*

(3) *$F \in \mathcal{P}_{\forall}(\mathcal{B}_{\forall}(L))$.*

Proof. (1) $\Rightarrow$ (2) Let $F \in \mathcal{B}(\mathcal{F}_{\forall}(L))$. Then there is $G \in \mathcal{F}_{\forall}(L)$ such that $F \cap G = \{1\}$ and $F \vee G = L$. From $F \cap G = \{1\}$, it follows that $G \subseteq F_{\forall}^{\perp}$. Thus $L = F \vee G \subseteq F \vee F_{\forall}^{\perp}$.

(2) $\Rightarrow$ (3) Let $F \vee F_{\forall}^{\perp} = L$. Then there are $a \in F$ and $b \in F_{\forall}^{\perp}$ such that $a \odot b = 0$, which implies that $\forall a \odot \forall b = 0$. From $F \cap F_{\forall}^{\perp} = \{1\}$, it follows that $\langle a \rangle_{\forall} \cap \langle b \rangle_{\forall} = \langle \forall a \vee \forall b \rangle_{\forall} = \{1\}$, so $\forall a \vee \forall b = 1$. Hence a is a $\forall$-complemented element. We now prove that $F = \langle a \rangle_{\forall}$. Since $a \in F$, we have $\langle a \rangle_{\forall} \subseteq F$. Conversely,

suppose that $x \in F$. Since $b \in F_\forall^\perp$, we have $\forall x \vee \forall b = 1$. So $\forall a = \forall a \odot (\forall x \vee \forall b) = (\forall a \odot \forall x) \vee (\forall a \odot \forall b) \le x$, which means that $x \in \langle a \rangle_\forall$. Thus $F = \langle a \rangle_\forall$.

(3) $\Rightarrow$ (1) Let $F \in \mathcal{P}_\forall(\mathcal{B}_\forall(L))$. Then there is a $\forall$-complemented element a such that $F = \langle a \rangle_\forall$. Since a is a $\forall$-complemented element, then there is $b \in L$ such that $\forall a \vee \forall b = 1$ and $\forall a \wedge \forall b = \forall a \odot \forall b = 0$. Applying Proposition 2.6(6), $F \cap \langle b \rangle_\forall = \langle a \rangle_\forall \cap \langle b \rangle_\forall = \langle \forall a \vee \forall b \rangle_\forall = \{1\}$. By Proposition 2.6(7), $F \vee \langle b \rangle_\forall = \langle a \rangle_\forall \vee \langle b \rangle_\forall = \langle \forall a \rangle_\forall \vee \langle \forall b \rangle_\forall = \langle \forall a \odot \forall b \rangle_\forall = \langle 0 \rangle_\forall = L$. Thus, F is a complemented element of $\mathcal{F}_\forall(L)$ as a lattice. □

Theorem 3.16. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then the following are equivalent:*

(1) $(\mathcal{F}_\forall(L), \cap, \vee, ^*, \{1\}, L)$ *is a Boolean algebra.*

(2) *every monadic filter of L is principal and for every $x \in L$, there is $n \ge 1$ such that $x \vee ((\forall x)^n)^* = 1$.*

Proof. (1) $\Rightarrow$ (2) Suppose that $F \in \mathcal{F}_\forall(L)$. Since $\mathcal{F}_\forall(L)$ is a Boolean algebra, then $F \vee F_\forall^\perp = L$. From Lemma 3.15 it follows that $F = \langle a \rangle_\forall$, for some monadic complement $a \in L$. This means that F is principal.

Since $\mathcal{F}_\forall(L)$ is a Boolean algebra, then $\langle x \rangle_\forall \vee x_\forall^\perp = L$, for all $x \in L$. So there are $c \in \langle x \rangle_\forall$ and $d \in x_\forall^\perp$ such that $c \odot d = 0$. Hence there is $n \ge 1$ such that $(\forall x)^n \odot d \le c \odot d = 0$, so $d \le ((\forall x)^n)^*$. Since $d \in x_\forall^\perp$, we have $\forall d \vee \forall x = 1$. Thus $1 = \forall d \vee \forall x \le ((\forall x)^n)^* \vee x$ and so $x \vee ((\forall x)^n)^* = 1$.

(2) $\Rightarrow$ (1) To prove that $\mathcal{F}_\forall(L)$ is a Boolean algebra. Since $\mathcal{F}_\forall(L)$ is a Heyting algebra, it is sufficient to show that for all $F \in \mathcal{F}_\forall(L)$, $F_\forall^\perp = \{1\}$ only if $F = L$. From the hypothesis it follows that $F = \langle a \rangle_\forall$, for some $a \in L$ and there is $n \ge 1$ such that $a \vee ((\forall a)^n)^* = 1$. By Proposition 2.4(8) and (4), we have $((\forall a)^n)^* \in a_\forall^\perp = \{1\}$. This implies that $(\forall a)^n \le ((\forall a)^n)^{**} = 1^* = 0$, that is, $0 \in \langle a \rangle_\forall = F$. We conclude that $F = L$. □

It is proved by Rasouli et al. [23] that the hull-kernel topology coincides with the stable topology on *Spec*(L) in a residuated lattice L if and if only L is hyperarchimedean. We now introduce hyperarchimedean monadic residuated lattices as a natural generalization of hyperarchimedean residuated lattices. Some conditions are give for a monadic residuated lattice to be hyperarchimedean.

Definition 3.17. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. An element $x \in L$ is called monadic archimedean if there is $n \in \mathbb{N}, n \ge 1$ such that $x \vee ((\forall x)^n)^* = 1$. A monadic residuated lattice $(L, \exists, \forall)$ is called hyperarchimedean if any element $x \in L$ is monadic archimedean.*

Example 3.18. *Let* $L = \{0, a, b, 1\}$ *and* $0 < a, b < 1$. *The operations* $\odot$ *and* $\to$ *on* L *are defined as follows:*

$\odot$	0	a	b	1
0	0	0	0	0
a	0	a	0	a
b	0	0	b	b
1	0	a	b	1

$\to$	0	a	b	1
0	1	1	1	1
a	b	1	b	1
b	a	a	1	1
1	0	a	b	1

The operations $\forall, \exists$ *are defined on* L *as follows:*

x	0	a	b	1
$\forall_1 x$	0	a	b	1
$\forall_2 x$	0	0	0	1

x	0	a	b	1
$\exists_1 x$	0	a	b	1
$\exists_2 x$	0	1	1	1

One can check that $(L, \forall_1, \exists_1)$ *and* $(L, \forall_2, \exists_2)$ *are hyperarchimedean monadic residuated lattices.*

Some necessary and sufficient conditions will be given for a monadic residuated lattice to be hyperarchimedean in the subsequent theorems.

Theorem 3.19. *Let* $(L, \exists, \forall)$ *be a monadic residuated lattice. Then the following are equivalent:*

(1) $\mathcal{B}(\mathcal{F}_\forall(L)) = \mathcal{P}_\forall(L)$*;*

(2) $\mathcal{P}_\forall(\mathcal{B}_\forall(L)) = \mathcal{P}_\forall(L)$*;*

(3) $(L, \exists, \forall)$ *is hyperarchimedean.*

Proof. (1) $\Leftrightarrow$ (2) It is trivial by Lemma 3.15.

(1) $\Rightarrow$ (3) Let $\mathcal{B}(\mathcal{F}_\forall(L)) = \mathcal{P}_\forall(L)$. For any $x \in L$, there is some $F \in \mathcal{B}(\mathcal{F}_\forall(L))$ such that $\langle x \rangle_\forall = F$. From Lemma 3.15 it follows that $F \vee F_\forall^\perp = L$, so $\langle x \rangle_\forall \vee x_\forall^\perp = L$. Hence there exist $a \in \langle x \rangle_\forall$ and $b \in x_\forall^\perp$ such that $a \odot b = 0$, which implies that $b \leq a^*$. From $a \in \langle x \rangle_\forall$, it follows that $a \geq (\forall x)^n$ for some $n \geq 1$ and so $a^* \leq ((\forall x)^n)^*$, which implies that $b \leq ((\forall x)^n)^*$. From $b \in x_\forall^\perp$, it follows that $\forall x \vee \forall b = 1$. This yields that $1 = \forall x \vee \forall b \leq x \vee ((\forall x)^n)^*$ for some $n \geq 1$. Therefore, $(L, \exists, \forall)$ is hyperarchimedean.

(3) $\Rightarrow$ (2) Let $(L, \exists, \forall)$ be hyperarchimedean. For any $x \in L$, there is $n \in \mathbb{N}, n \geq 1$ such that $x \vee ((\forall x)^n)^* = 1$, so $\forall x \vee \forall((\forall x)^n)^* = 1$. Applying Proposition 2.6(4) and (6), $\langle x \rangle_\forall \cap \langle ((\forall x)^n)^* \rangle_\forall = \langle \forall x \vee \forall((\forall x)^n)^* \rangle_\forall = \{1\}$. Applying Proposition 2.6(4) and (7), $\langle x \rangle_\forall \vee \langle ((\forall x)^n)^* \rangle_\forall = \langle \forall x \rangle_\forall \vee \langle ((\forall x)^n)^* \rangle_\forall = \langle (\forall x)^n \rangle_\forall \vee \langle ((\forall x)^n)^* \rangle_\forall = \langle (\forall x)^n \odot ((\forall x)^n)^* \rangle_\forall = \langle 0 \rangle_\forall = L$. Thus, $\mathcal{P}_\forall(\mathcal{B}_\forall(L)) = \mathcal{P}_\forall(L)$. $\square$

Theorem 3.20. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then the following are equivalent:*

(1) *$(L, \exists, \forall)$ is hyperarchimedean;*

(2) *$Max_\forall(L) = Spec_\forall(L)$;*

(3) *$Spec_\forall(L) = Min_\forall(L)$.*

Proof. (1) $\Rightarrow$ (2) It is obvious that $Max_\forall(L) \subseteq Spec_\forall(L)$. Let $P \in Spec_\forall(L)$ and $x \in L \setminus P$. Since $(L, \exists, \forall)$ is hyperarchimedean, then x is monadic archimedean and so there is $n \geq 1$ such that $x \vee ((\forall x)^n)^* = 1$. Applying Proposition 2.4(1) and (8), $\forall x \vee \forall(((\forall x)^n)^*) = \forall 1 = 1$. This means that $((\forall x)^n)^* \in P$ as P is prime. From Proposition 2.10(4) it follows that $P \in Max_\forall(L)$. Thus $Spec_\forall(L) \subseteq Max_\forall(L)$.

(2) $\Rightarrow$ (3) Suppose that $Q \in Spec_\forall(L)$ such that $Q \subseteq P$, for any $P \in Spec_\forall(L)$. Since $Max_\forall(L) = Spec_\forall(L)$, then $Q \in Max_\forall(L)$, so $Q = P$. This means that P is a minimal prime monadic filter.

(3) $\Rightarrow$ (1) Let $x \in L$ such that $x \neq 1$. Since $x \in \langle x \rangle_\forall$ and $x \neq 1$, by Lemma 3.4(11) we have $x \notin x^{\perp}_{\forall}$. Set $F = \langle x^{\perp}_{\forall}, x \rangle_\forall$ the monadic filter generated by $x^{\perp}_{\forall}$ and x . We will prove that $F = L$. Suppose that $F \neq L$. Then there is $P \in Spec_\forall(L)$ such that $F \subseteq P$. Since P is a minimal prime monadic filter and $x \in F \subseteq P$, by Corollary 3.14(3) we have $x^{\perp}_{\forall} \nsubseteq P$, a contradiction. Hence $F = L$ and so there is $n \geq 1$ and $f \in x^{\perp}_{\forall}$ such that $f \odot (\forall x)^n = 0$, which implies that $f \leq ((\forall x)^n)^*$. From $f \in x^{\perp}_{\forall}$, we have $\forall x \vee \forall f = 1$. Thus $x \vee ((\forall x)^n)^* \geq \forall x \vee f \geq \forall x \vee \forall f = 1$. Therefore, $(L, \exists, \forall)$ is hyperarchimedean. □

4 Monadic α-filters

In this section, the notion of monadic α-filters in monadic residuated lattices is presented by means of monadic co-annihilators. Some basic and important properties of monadic α-filters are obtained. The lattice structure of all monadic α-filters is investigated and prime monadic α-filter theorems are established.

Definition 4.1. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. A monadic filter F of $(L, \exists, \forall)$ is called a monadic α-filter if $(x^{\perp}_{\forall})^{\perp}_{\forall} \subseteq F$, for all $x \in F$.*

The set of all monadic α-filters of a monadic residuated lattice $(L, \exists, \forall)$ is denoted by $\mathcal{F}^{\alpha}_{\forall}(L)$. It is obvious that $(L^{\perp}_{\forall})^{\perp}_{\forall} = L$ and $(1^{\perp}_{\forall})^{\perp}_{\forall} = \{1\}$. Hence both $\{1\}$ and L are monadic α-filters of $(L, \exists, \forall)$.

Example 4.2. (1) *Consider the monadic residuated lattice* $(L,\exists,\forall)$ *in Example 3.7(2). Routine calculation shows that* $(c_\forall^\perp)_\forall^\perp = \{c,1\} = F_1$, $(d_\forall^\perp)_\forall^\perp = \{d,1\} = F_2$, $(b_\forall^\perp)_\forall^\perp = \{1\}_\forall^\perp = L$ *and* $(1_\forall^\perp)_\forall^\perp = \{1\}$. *Thus, both* F_1 *and* F_2 *are monadic* α*-filters, but* $F_3 = \{b,c,d,1\}$ *is not a monadic* α*-filter as* $(b_\forall^\perp)_\forall^\perp = L \nsubseteq F_3$.

(2) *Consider the monadic residuated lattices* $(L,\forall_1,\exists_1)$ *and* $(L,\forall_2,\exists_2)$ *in Example 3.2. Clearly, it is easily checked that* $A_{\forall_1}^\perp = \{1\}$ *and* $A_{\forall_2}^\perp = \{1\}$, *for any nonempty subset* $A \neq \{1\}$ *of* L. *Therefore,* $\{1\}$ *and* L *are the only monadic* α*-filters in monadic residuated lattices* $(L,\forall_1,\exists_1)$ *and* $(L,\forall_2,\exists_2)$.

However, in the residuated lattice L *routine calculation shows that* $\{b,1\}^\perp = \{c,1\}, \{c,1\}^\perp = \{b,1\}$ *and consequently* $\{b,1\}^{\perp\perp} = \{b,1\}, \{c,1\}^{\perp\perp} = \{c,1\}$, *where* $^\perp$ *stands for the co-annihilator in the residuated lattice* L. *This means that both* $\{b,1\}$ *and* $\{c,1\}$ *are* α*-filters of* L, *but* $\{b,1\}$ *and* $\{c,1\}$ *are not monadic* α*-filters of the monadic residuated lattice* $(L,\forall_1,\exists_1)$ *and* $(L,\forall_2,\exists_2)$.

Notice that the aforementioned Example 4.2(2) indicates that the notion of monadic α-filters in a monadic residuated lattice $(L,\exists,\forall)$ is different from the notion of α-filters in the residuated lattice L in general and the former is more general than the latter. However, the notion of monadic α-filters in a monadic residuated lattice $(L,\exists,\forall)$ coincides with the notion of α-filters in the residuated lattice L whenever the two operators $\exists : L \to L$ and $\forall : L \to L$ are identity maps.

Let $(L,\exists,\forall)$ be a monadic residuated lattice and F be a monadic filter of $(L,\exists,\forall)$. The monadic α-filter of $(L,\exists,\forall)$ generated by F is denoted by $\alpha_\forall(F)$ and it is obvious that $\mathcal{F}_\forall^\alpha(L) = \{\alpha_\forall(F) \mid F \in \mathcal{F}_\forall(L)\}$. The following proposition indicates a concrete description of $\alpha_\forall(F)$.

Proposition 4.3. *Let* $(L,\exists,\forall)$ *be a monadic residuated lattice and* F *be a monadic filter of* $(L,\exists,\forall)$. *Then* $\alpha_\forall(F) = \{x \in L \mid a_\forall^\perp \subseteq x_\forall^\perp, for\ some\ a \in F\}$.

Proof. Set $H_F = \{x \in L \mid a_\forall^\perp \subseteq x_\forall^\perp, for\ some\ a \in F\}$. Firstly, it is easily proved that H_F is a monadic filter of $(L,\exists,\forall)$ and $F \subseteq H_F$. Let $x,y \in H_F$. Then there exist $a,b \in F$ such that $a_\forall^\perp \subseteq x_\forall^\perp$ and $b_\forall^\perp \subseteq y_\forall^\perp$, so $a_\forall^\perp \cap b_\forall^\perp \subseteq x_\forall^\perp \cap y_\forall^\perp$. By Lemma 3.4(6) we have $(a \odot b)_\forall^\perp \subseteq (x \odot y)_\forall^\perp$, which means that $x \odot y \in H_F$. If $x \in \alpha_\forall(F)$ and $x \leq y$ for all $x,y \in L$, then there is $a \in F$ such that $a_\forall^\perp \subseteq x_\forall^\perp$. By Lemma 3.4(1) and $x \leq y$, we have $x_\forall^\perp \subseteq y_\forall^\perp$. Hence $a_\forall^\perp \subseteq y_\forall^\perp$ and so $y \in H_F$. If $x \in H_F$ for all $x \in L$, then there is $a \in F$ such that $a_\forall^\perp \subseteq x_\forall^\perp$. By Lemma 3.4(5), $x_\forall^\perp = (\forall x)_\forall^\perp$ and so $a_\forall^\perp \subseteq (\forall x)_\forall^\perp$. This means that $\forall x \in H_F$. We conclude that H_F is a monadic filter of $(L,\exists,\forall)$. Obviously, $F \subseteq H_F$.

Next, we will prove that H_F is a monadic α-filter. If $x \in H_F$, then $a_\forall^\perp \subseteq x_\forall^\perp$ for some $a \in F$. Let $z \in (x_\forall^\perp)_\forall^\perp$. Then by Lemma 3.4(3) and (4), $x_\forall^\perp = ((x_\forall^\perp)_\forall^\perp)_\forall^\perp \subseteq z_\forall^\perp$. It follows that $a_\forall^\perp \subseteq z_\forall^\perp$ and so $z \in H_F$. This implies that $(x_\forall^\perp)_\forall^\perp \subseteq H_F$.

Finally, suppose that G is a monadic α-filter such that $F \subseteq G$. If $x \in H_F$, then $a_{\forall}^{\perp} \subseteq x_{\forall}^{\perp}$ for some $a \in F$. By Lemma 3.4(3) and (1), $x \in (x_{\forall}^{\perp})_{\forall}^{\perp} \subseteq (a_{\forall}^{\perp})_{\forall}^{\perp} \subseteq G$. Thus $H_F \subseteq G$.

We conclude that H_F is the smallest monadic α-filter containing F. $\square$

Let $(L, \exists, \forall)$ be a monadic residuated lattice. The monadic α-filter generated by a subset X of L is denoted by $\alpha_{\forall}(X)$ and it is clear that $\alpha_{\forall}(X) = \alpha_{\forall}(\langle X \rangle_{\forall})$. The following proposition provides some basic properties of the generated monadic α-filters.

Proposition 4.4. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F, G be monadic filters of $(L, \exists, \forall)$. Then the following hold for $X \subseteq L$ and $x, y \in L$:*

(1) $\alpha_{\forall}(F) = \{x \in L \mid (x_{\forall}^{\perp})_{\forall}^{\perp} \subseteq (a_{\forall}^{\perp})_{\forall}^{\perp}, for\ some\ a \in F\} = \bigcup_{a \in F} (a_{\forall}^{\perp})_{\forall}^{\perp}$;

(2) $\alpha_{\forall}(X) = \{a \in L \mid a \in ((\forall x_1 \odot \cdots \odot \forall x_n)_{\forall}^{\perp})_{\forall}^{\perp}, for\ some\ x_1, \cdots, x_n \in X\}$;

(3) *$F \subseteq G$ implies $\alpha_{\forall}(F) \subseteq \alpha_{\forall}(G)$;*

(4) $\alpha_{\forall}(F) \cap \alpha_{\forall}(G) = \alpha_{\forall}(F \cap G)$;

(5) $\alpha_{\forall}(x) \cap \alpha_{\forall}(y) = \alpha_{\forall}(\forall x \vee \forall y)$;

(6) $\alpha_{\forall}(F) \vee \alpha_{\forall}(G) \subseteq \alpha_{\forall}(F \vee G)$;

(7) $\alpha_{\forall}(F, x) = \bigcup_{f \in F, n \geq 1} ((f \odot (\forall x)^n)_{\forall}^{\perp})_{\forall}^{\perp}$;

(8) $\alpha_{\forall}(F, x) \cap \alpha_{\forall}(F, y) = \alpha_{\forall}(F, \forall x \vee \forall y)$;

(9) $\alpha_{\forall}(\langle F, x \rangle_{\forall} \vee \langle F, y \rangle_{\forall}) = \alpha_{\forall}(F, x \odot y)$;

(10) *if $\{F_i \mid i \in I\}$ is a family of monadic α-filters, then $\bigcap_{i \in I} F_i$ is a monadic α-filter;*

(11) *if $\{F_i \mid i \in I\}$ is a family of monadic filters, then $\alpha_{\forall}(\bigvee_{i \in I} F_i) = \{x \in L \mid x \in ((a_{i_1} \odot a_{i_2} \odot \cdots \odot a_{i_n})_{\forall}^{\perp})_{\forall}^{\perp}, for\ some\ a_{i_j} \in F_{i_j}, 1 \leq j \leq n, n \geq 1\}$.*

Proof. (1) It follows from Proposition 4.3 and Proposition 3.3(5).

(2) Set $A_X = \{a \in L \mid a \in ((\forall x_1 \odot \cdots \odot \forall x_n)_{\forall}^{\perp})_{\forall}^{\perp}, for\ some\ x_1, \cdots, x_n \in X\}$. Suppose that $a \in \alpha_{\forall}(X)$. Then $a \in \alpha_{\forall}(\langle X \rangle_{\forall})$. By (1), $(a_{\forall}^{\perp})_{\forall}^{\perp} \subseteq (b_{\forall}^{\perp})_{\forall}^{\perp}$, for some $b \in \langle X \rangle_{\forall}$. Hence there are $x_1, \cdots, x_n \in X$ such that $b \geq \forall x_1 \odot \cdots \odot \forall x_n$, for some $n \geq 1$. Applying Lemma 3.4(1), we have $(b_{\forall}^{\perp})_{\forall}^{\perp} \subseteq ((\forall x_1 \odot \cdots \odot \forall x_n)_{\forall}^{\perp})_{\forall}^{\perp}$. So $(a_{\forall}^{\perp})_{\forall}^{\perp} \subseteq ((\forall x_1 \odot \cdots \odot \forall x_n)_{\forall}^{\perp})_{\forall}^{\perp}$. By Lemma 3.4(3), $a \in A_X$. Conversely, let $a \in A_X$. Then there are $x_1, \cdots, x_n \in X$ such that $a \in ((\forall x_1 \odot \cdots \odot \forall x_n)_{\forall}^{\perp})_{\forall}^{\perp}$. This means that $a \in \alpha_{\forall}(\langle X \rangle_{\forall}) = \alpha_{\forall}(X)$.

(3) Let $x \in \alpha_\forall(F)$. Then $(x_\forall^\perp)_\forall^\perp \subseteq (a_\forall^\perp)_\forall^\perp$, for some $a \in F$. From $F \subseteq G$, it follows that $(x_\forall^\perp)_\forall^\perp \subseteq (a_\forall^\perp)_\forall^\perp$, for some $a \in G$. Thus $\alpha_\forall(F) \subseteq \alpha_\forall(G)$.

(4) Since $F \cap G \subseteq F, G$, By (3) we have $\alpha_\forall(F \cap G) \subseteq \alpha_\forall(F) \cap \alpha_\forall(G)$. Conversely, let $x \in \alpha_\forall(F) \cap \alpha_\forall(G)$. Then there are $f \in F$ and $g \in G$ such that $(x_\forall^\perp)_\forall^\perp \subseteq (f_\forall^\perp)_\forall^\perp$ and $(x_\forall^\perp)_\forall^\perp \subseteq (g_\forall^\perp)_\forall^\perp$. By Lemma 3.4(8), we have $(x_\forall^\perp)_\forall^\perp \subseteq (f_\forall^\perp)_\forall^\perp \cap (g_\forall^\perp)_\forall^\perp = ((\forall f \vee \forall g)_\forall^\perp)_\forall^\perp$, for $\forall f \vee \forall g \in F \cap G$, which means that $x \in \alpha_\forall(F \cap G)$. Thus $\alpha_\forall(F) \cap \alpha_\forall(G) \subseteq \alpha_\forall(F \cap G)$. We conclude that $\alpha_\forall(F) \cap \alpha_\forall(G) = \alpha_\forall(F \cap G)$

(5) Applying (4) and Proposition 2.6(6), we obtain successively that $\alpha_\forall(x) \cap \alpha_\forall(y) = \alpha_\forall(\langle x \rangle_\forall) \cap \alpha_\forall(\langle y \rangle_\forall) = \alpha_\forall(\langle x \rangle_\forall \cap \langle y \rangle_\forall) = \alpha_\forall(\langle \forall x \vee \forall y \rangle_\forall) = \alpha_\forall(\forall x \vee \forall y)$.

(6) Since $F, G \subseteq F \vee G$, by (3) we have $\alpha_\forall(F) \vee \alpha_\forall(G) \subseteq \alpha_\forall(F \vee G)$.

(7) By (2) and Proposition 2.6(3), we have $\alpha_\forall(F, x) = \bigcup_{f \in F, n \geq 1}((f \odot (\forall x)^n)_\forall^\perp)_\forall^\perp$.

(8) Applying (4) and Lemma 2.7(2), we have the following formulas: $\alpha_\forall(F, x) \cap \alpha_\forall(F, y) = \alpha_\forall(\langle F, x \rangle_\forall \cap \langle F, y \rangle_\forall) = \alpha_\forall(\langle F, \forall x \vee \forall y \rangle_\forall) = \alpha_\forall(F, \forall x \vee \forall y)$.

(9) According to Lemma 2.7(3), we have the following formulas: $\alpha_\forall(\langle F, x \rangle_\forall \vee \langle F, y \rangle_\forall) = \alpha_\forall(\langle F, x \odot y) \rangle_\forall) = \alpha_\forall(F, x \odot y)$.

(10) Let $\{F_i \mid i \in I\}$ be a family of monadic α-filters and $x \in \bigcap_{i \in I} F_i$. Then $x \in F_i$ for all $i \in I$. Since F_i is a monadic α-filter for all $i \in I$, we have $(x_\forall^\perp)_\forall^\perp \subseteq F_i$ for all $i \in I$, so $(x_\forall^\perp)_\forall^\perp \subseteq \bigcap_{i \in I} F_i$.

(11) This follows from (1) and Lemma 2.7(5). □

Proposition 4.5. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then:*

(1) *$A_\forall^\perp$ is a monadic α-filter of $(L, \exists, \forall)$, for a subset A of L;*

(2) *any minimal prime monadic filter is a monadic α-filter.*

Proof. (1) Let $x \in A_\forall^\perp$. Then $(x_\forall^\perp)_\forall^\perp \subseteq ((A_\forall^\perp)_\forall^\perp)_\forall^\perp = A_\forall^\perp$ by Proposition 3.3(2) and (4). Thus $A_\forall^\perp$ is a monadic α-filter.

(2) Let M be a minimal prime monadic filter of (L, σ). Assume that $x \in M$. Applying Corollary 3.14(2), we have $x \in D_\forall(M)$. So there is $y \in L \setminus M$ such that $x \in y_\forall^\perp$. By Lemma 3.4(3) and (4), we have $(x_\forall^\perp)_\forall^\perp \subseteq ((y_\forall^\perp)_\forall^\perp)_\forall^\perp = y_\forall^\perp$. Since $y \in L \setminus M$, by Corollary 3.14(3) we have $y_\forall^\perp \subseteq M$. Hence $(x_\forall^\perp)_\forall^\perp \subseteq M$. This states that M is a monadic α-filter. □

Proposition 4.6. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. Then the following are equivalent:*

(1) *F is a monadic α-filter of $(L, \exists, \forall)$;*

(2) *$\alpha_\forall(F) = F$;*

(3) *$x_\forall^\perp = y_\forall^\perp$ and $x \in F$ imply $y \in F$, for all $x, y \in L$;*

(4) $F = \bigcup_{x\in F}(x_{\forall}^{\perp})_{\forall}^{\perp}$.

Proof. (1) $\Leftrightarrow$ (2) It is obvious.

(2) $\Rightarrow$ (3) Suppose that $x_{\forall}^{\perp} = y_{\forall}^{\perp}$ and $x \in F$, for $x, y \in L$. It follows from Proposition 4.3 that $y \in \alpha_{\forall}(F)$. By (2), we have $y \in F$.

(3) $\Rightarrow$ (2) Let $x \in \alpha_{\forall}(F)$. Then $(x_{\forall}^{\perp})_{\forall}^{\perp} \subseteq (a_{\forall}^{\perp})_{\forall}^{\perp}$, for some $a \in F$. By Lemma 3.4(8), we have $(x_{\forall}^{\perp})_{\forall}^{\perp} = (x_{\forall}^{\perp})_{\forall}^{\perp} \cap (a_{\forall}^{\perp})_{\forall}^{\perp} = ((\forall x \vee \forall a)_{\forall}^{\perp})_{\forall}^{\perp}$. Applying Lemma 3.4(3), $x_{\forall}^{\perp} = (\forall x \vee \forall a)_{\forall}^{\perp}$. Since F is a monadic filter and $a \in F$, it follows that $\forall x \vee \forall a \in F$. By the hypothesis we have $x \in F$.

(1) $\Leftrightarrow$ (4) Let F be an monadic α-filter of $(L, \exists, \forall)$. Applying Lemma 3.4(3), we have $F = \bigcup_{x\in F}\{x\} \subseteq \bigcup_{x\in F}(x_{\forall}^{\perp})_{\forall}^{\perp} \subseteq F$. Thus $F = \bigcup_{x\in F}(x_{\forall}^{\perp})_{\forall}^{\perp}$. The converse is evident. □

Theorem 4.7. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. The map $\Phi : \mathcal{F}_{\forall}(L) \to \mathcal{F}(\mathcal{L}_{\forall}^{\perp})$, defined by $\Phi(F) = \{x_{\forall}^{\perp} \mid x \in F\}$ and the map $\Psi : \mathcal{F}(\mathcal{L}_{\forall}^{\perp}) \to \mathcal{F}_{\forall}(L)$, defined by $\Psi(\mathcal{F}) = \{x \in L \mid x_{\forall}^{\perp} \in \mathcal{F}\}$, respectively, where $\mathcal{F}(\mathcal{L}_{\forall}^{\perp})$ denotes the set of all filters of $\mathcal{L}_{\forall}^{\perp}$ as a lattice. Then:*

(1) *The pair (Φ, Ψ) is an adjunction and so $\Psi \circ \Phi$ is a closure operator on the lattice $\mathcal{F}_{\forall}(L)$;*

(2) *$\Psi(\Phi(F)) = \alpha(F)$, for all $F \in \mathcal{F}_{\forall}(L)$;*

(3) *F is a monadic α-filter of $(L, \exists, \forall)$ if and only if $\Psi(\Phi(F)) = F$, that is, F is precisely a closed element with respect to $\Psi \circ \Phi$ on $\mathcal{F}_{\forall}(L)$.*

Proof. (1) Assume that $\Phi(F) \subseteq \mathcal{F}$. If $x \in F$, then $x_{\forall}^{\perp} \in \Phi(F)$ and so $x_{\forall}^{\perp} \in \mathcal{F}$, which means that $F \subseteq \Psi(\mathcal{F})$. Conversely, suppose that $F \subseteq \Psi(\mathcal{F})$. Consider $u \in \Phi(F)$. Then $u = a_{\forall}^{\perp}$ for some $a \in F$ and so $a \in \Psi(\mathcal{F})$. Thus $a_{\forall}^{\perp} \in \mathcal{F}$, that is, $u \in \mathcal{F}$. This shows that $\Phi(F) \subseteq \mathcal{F}$. Therefore, the pair (Φ, Ψ) is an adjunction and consequently $\Psi \circ \Phi$ is a closure operator on $\mathcal{F}_{\forall}(L)$.

(2) Let F be a monadic filter of $(L, \exists, \forall)$. Then $\Psi(\Phi(F)) = \{x \in L \mid x_{\forall}^{\perp} \in \Phi(F)\} = \{x \in L \mid x_{\forall}^{\perp} = a_{\forall}^{\perp},\ for\ some\ a \in F\} = \{x \in L \mid x \in (a_{\forall}^{\perp})_{\forall}^{\perp},\ for\ some\ a \in F\} = \bigcup_{x\in F}(x_{\forall}^{\perp})_{\forall}^{\perp} = \alpha(F)$.

(3) This follows from (2) and Proposition 4.6(2). □

In a monadic residuated lattice $(L, \exists, \forall)$, an element $x \in L$ is called monadic dense if $x_{\forall}^{\perp} = \{1\}$. In Example 3.2, every element $1 \neq x \in L$ is monadic dense in $(L, \exists_1, \forall_1)$ and $(L, \exists_2, \forall_2)$.

Proposition 4.8. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic filter of $(L, \exists, \forall)$. Then:*

(1) *$\alpha_\forall(F) = L$ if and only if there exists $x \in F$ such that $x_\forall^\perp = \{1\}$, that is, F contains a monadic dense element;*

(2) *$\alpha_\forall(F)$ is proper if and only if F has no monadic dense element;*

(3) *if F is a maximal monadic filter of $(L, \exists, \forall)$ such that $\alpha_\forall(F) \neq L$, then F is a monadic α-filter of $(L, \exists, \forall)$.*

Proof. (1) If $\alpha_\forall(F) = L$, then $0 \in \alpha_\forall(F)$. Hence there is $x \in F$ such that $x_\forall^\perp \subseteq 0_\forall^\perp = \{1\}$ and so $x_\forall^\perp = \{1\}$. Conversely, suppose that there exists $x \in F$ such that $x_\forall^\perp = \{1\}$. Since $0_\forall^\perp = \{1\}$, we have $x_\forall^\perp = 0_\forall^\perp$. From Proposition 4.3 it follows that $0 \in \alpha_\forall(F)$. Thus, $\alpha_\forall(F) = L$.

(2) This follows from (1).

(3) It is obvious that $F \subseteq \alpha_\forall(F) \subseteq L$. Since F is a maximal monadic filter and $\alpha_\forall(F) \neq L$, we have $\alpha_\forall(F) = F$. We conclude that F is a monadic α-filter of $(L, \exists, \forall)$. □

A frame L is called coherent ([24]) if the following conditions are satisfied: (1) The set $K(L)$ of all compact elements of L is a sublattice of L; (2) any element of L is a join of compact elements of L.

Let $\{F_i \mid i \in I\}$ be a family of monadic filters of a monadic residuated lattice $(L, \exists, \forall)$. We define a join operation $\bigvee_\alpha$ on $\mathcal{F}_\forall^\alpha(L)$: $\bigvee_\alpha\{\alpha_\forall(F_i) \mid i \in I\} = \alpha_\forall(\bigvee_{i\in I} F_i)$. The following theorem shows that the complete lattice $\mathcal{F}_\forall^\alpha(L)$ under the two binary operations $\bigcap$ and $\bigvee_\alpha$ forms a coherent frame.

Theorem 4.9. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then $(\mathcal{F}_\forall^\alpha(L), \bigcap, \bigvee_\alpha, \{1\}, L)$ is a coherent frame.*

Proof. (1) We prove that $(\mathcal{F}_\forall^\alpha(L), \bigcap, \bigvee_\alpha, \{1\}, L)$ is a frame. Suppose that F is a monadic filter and $\{F_i \mid i \in I\}$ is a family of monadic filters. From Proposition 4.4(4) and Proposition 2.9(1) it follows that $\alpha_\forall(F) \bigcap(\bigvee_\alpha\{\alpha_\forall(F_i) \mid i \in I\}) = \alpha_\forall(F) \bigcap \alpha_\forall(\bigvee_{i\in I} F_i) = \alpha_\forall(F \bigcap(\bigvee_{i\in I} F_i)) = \alpha_\forall(\bigvee_{i\in I}(F \bigcap F_i)) = \bigvee_\alpha\{\alpha_\forall(F \bigcap F_i) \mid i \in I\} = \bigvee_\alpha\{\alpha_\forall(F) \bigcap \alpha_\forall(F_i) \mid i \in I\}$. This shows that $(\mathcal{F}_\forall^\alpha(L), \bigcap, \bigvee_\alpha, \{1\}, L)$ is a frame.

(2) Now we verify that $(\mathcal{F}_\forall^\alpha(L), \bigcap, \bigvee_\alpha, \{1\}, L)$ is coherent.

Firstly, from Proposition 2.9(2) it follows that $\langle x\rangle_\forall$ is compact in the lattice $\mathcal{F}_\forall(L)$. By the definition of $\bigvee_\alpha$ on the lattice $\mathcal{F}_\forall^\alpha(L)$, we deduce that $\alpha_\forall(x)$ is compact in $\mathcal{F}_\forall^\alpha(L)$.

Next, by $K(\mathcal{F}_\forall^\alpha(L))$ we denote the set of all compact elements of $\mathcal{F}_\forall^\alpha(L)$ and $K(\mathcal{F}_\forall^\alpha(L))) = \{\alpha_\forall(x) \mid x \in L\}$. By Proposition 2.6(7) and Lemma 3.4(5), we have $\alpha_\forall(x) \bigvee_\alpha \alpha_\forall(y) = \alpha_\forall(\langle x\rangle_\forall \vee \langle y\rangle_\forall) = \alpha_\forall(\langle x \odot y\rangle_\forall) = \alpha_\forall(x \odot y)$, for all $x, y \in L$. By

Proposition 4.4(5), we have $\alpha_\forall(x) \bigcap \alpha_\forall(y) = \alpha_\forall(\forall x \vee \forall y)$. Therefore, $K(\mathcal{F}^\alpha_\forall(L))$ is a sublattice of $\mathcal{F}^\alpha_\forall(L)$.

Finally, since $F = \vee_{x\in F}\langle x\rangle_\forall$ for any $F \in \mathcal{F}_\forall(L)$, by Proposition 4.4(11) we have $\alpha_\forall(F) = \alpha_\forall(\vee_{x\in F}\langle x\rangle_\forall) = \bigvee_\alpha\{\alpha_\forall(x) \mid x \in F\}$.

Combining (1) and (2), we conclude that $\mathcal{F}^\alpha_\forall(L)$ is a coherent frame. □

Theorem 4.10. *Let $(L, \exists, \forall)$ be a monadic residuated lattice. Then $(\mathcal{F}^\alpha_\forall(L), \cap, \vee_\alpha, {}^*, \{1\}, L)$ is a Boolean algebra if and only if for all $F \in \mathcal{F}_\forall(L)$, there exist $x \in F$ and $y \in F^\perp_\forall$ such that $x^\perp_\forall \cap y^\perp_\forall = \{1\}$, where $(\alpha_\forall(F))^* = F^\perp_\forall$.*

Proof. Suppose that $(\mathcal{F}^\alpha_\forall(L), \cap, \vee_\alpha, {}^*, \{1\}, L)$ is a Boolean algebra. Let $F \in \mathcal{F}_\forall(L)$. Then $L = \alpha_\forall(F) \vee_\alpha (\alpha_\forall(F))^* = \alpha_\forall(F) \vee_\alpha \alpha_\forall(F^\perp_\forall) = \alpha_\forall(F \vee F^\perp_\forall)$. Applying Proposition 4.8(1), there exists $a \in F \vee F^\perp_\forall$ such that $a^\perp_\forall = \{1\}$. By $a \in F \vee F^\perp_\forall$, we have $a \geq x \odot y$, for some $x \in F$ and $y \in F^\perp_\forall$. By Lemma 3.4(6) and (1), we have $x^\perp_\forall \cap y^\perp_\forall = (x \odot y)^\perp_\forall \subseteq a^\perp_\forall = \{1\}$ and so $x^\perp_\forall \cap y^\perp_\forall = \{1\}$. Conversely, assume that $F \in \mathcal{F}_\forall(L)$. By the hypothesis there are $x \in F$ and $y \in F^\perp_\forall$ such that $x^\perp_\forall \cap y^\perp_\forall = \{1\}$. By Lemma 3.4(6), we have $(x \odot y)^\perp_\forall = x^\perp_\forall \cap y^\perp_\forall = \{1\}$. It follows from Proposition 4.8(1) that $\alpha_\forall(F \vee F^\perp_\forall) = L$ as $x \odot y \in F \vee F^\perp_\forall$. Thus $\alpha_\forall(F) \vee_\alpha \alpha_\forall(F^\perp_\forall) = L$. Clearly, $\alpha_\forall(F) \cap \alpha_\forall(F^\perp_\forall) = \alpha_\forall(F \cap F^\perp_\forall) = \alpha_\forall(\{1\}) = \{1\}$. This shows that $\alpha_\forall(F^\perp_\forall)$ is the complement of $\alpha_\forall(F)$. Therefore, $(\mathcal{F}^\alpha_\forall(L), \cap, \vee_\alpha, {}^*, \{1\}, L)$ is a Boolean algebra. □

Theorem 4.11. (Prime monadic α-filter theorem) *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F be a monadic α-filter of $(L, \exists, \forall)$. If S is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ such that $F \cap S = \emptyset$, then there exists a prime monadic α-filter P of $(L, \exists, \forall)$ such that $F \subseteq P$ and $P \cap S = \emptyset$.*

Proof. Denote $\mathcal{F}_F = \{G \in \mathcal{F}^\alpha_\forall(L) \mid F \subseteq G \text{ and } G \cap S = \emptyset\}$. Since $F \in \mathcal{F}_F$, then $\mathcal{F}_F \neq \emptyset$. It is easily proved that $\mathcal{F}_F$ is a partially order set under the inclusion relation $\subseteq$. If $\{G_i \mid i \in I\}$ is a chain in $\mathcal{F}_F$, then $\bigcup_{i\in I} G_i$ is a monadic α-filter of $(L, \exists, \forall)$ and it is an upper bound of this chain. By Zorn's Lemma, there is a maximal element P in $\mathcal{F}_F$. We shall prove that P is a prime monadic α-filter of $(L, \exists, \forall)$. Suppose that $\forall x \vee \forall y \in P$ and neither $x \notin P$ or $y \notin P$. Construct the sets $\alpha_\forall(P, x)$ and $\alpha_\forall(P, y)$. From the maximality of P it follows that $\alpha_\forall(P, x) \notin \mathcal{F}_F$ and $\alpha_\forall(P, y) \notin \mathcal{F}_F$, which means that $\alpha_\forall(P, x) \cap S \neq \emptyset$ and $\alpha_\forall(P, y) \cap S \neq \emptyset$. Let $a \in \alpha_\forall(P, x) \cap S$ and $b \in \alpha_\forall(P, y) \cap S$, for some $a, b \in L$. Then $\forall a \vee \forall b \in S$ and $\forall a \vee \forall b \in \alpha_\forall(P, x) \cap \alpha_\forall(P, y) = \alpha_\forall(P, \forall x \vee \forall y)$ by Proposition 4.4(8). Since $\forall x \vee \forall y \in P$, we have $\alpha_\forall(P, \forall x \vee \forall y) = \alpha_\forall(P)$ and so $\forall a \vee \forall b \in \alpha_\forall(P) = P$. Hence $\forall a \vee \forall b \in P \cap S$, a contradiction. This states that P is a prime monadic α-filter of $(L, \exists, \forall)$. □

We shall denote by $Spec^{\alpha}_{\forall}(L)$ the set of all prime monadic α-filters of $(L, \exists, \forall)$ and $Spec^{\alpha}_{\forall}(L) = Spec_{\forall}(L) \cap \mathcal{F}^{\alpha}_{\forall}(L)$. The following proposition is evident by Theorem 4.11.

Proposition 4.12. *Let $(L, \exists, \forall)$ be a monadic residuated lattice and F an monadic α-filter of $(L, \exists, \forall)$. Then the following hold for all $X \subseteq L$:*

(1) *If $a \notin F$, then there is $P \in Spec^{\alpha}_{\forall}(L)$ such that $F \subseteq P$ and $a \notin P$;*

(2) *if $X \nsubseteq F$, then there is $P \in Spec^{\alpha}_{\forall}(L)$ such that $F \subseteq P$ and $X \nsubseteq P$;*

(3) *$\alpha_{\forall}(X) = \bigcap\{P \in Spec^{\alpha}_{\forall}(L) \mid X \subseteq P\}$;*

(4) *$\bigcap Spec^{\alpha}_{\forall}(L) = \{1\}$.*

Proof. (1) Set $S_a = \{x \in L \mid \forall x \leq \forall a\}$, for $a \in L$. It is clear that S_a is a $\forall$-$\vee$-closed subset of $(L, \exists, \forall)$ and $S_a \cap F = \emptyset$. From Theorem 4.11 it follows that there is a prime monadic α-filter P of $(L, \exists, \forall)$ such that $F \subseteq P$ and $a \notin P$.

(2) If $X \nsubseteq F$, then there is $a \in X$ and $a \notin F$. From (1) it follows that there is a prime monadic α-filter P of $(L, \exists, \forall)$ such that $a \notin P$. Thus $X \nsubseteq P$.

(3) Obviously, $\alpha_{\forall}(X) \subseteq \bigcap\{P \in Spec^{\alpha}_{\forall}(L) \mid X \subseteq P\}$. Conversely, suppose that $a \notin \alpha_{\forall}(X)$. By (1), there is a prime monadic α-filter P of $(L, \exists, \forall)$ such that $\alpha_{\forall}(X) \subseteq P$ and $a \notin P$. So $a \notin \bigcap\{P \in Spec^{\alpha}_{\forall}(L) \mid X \subseteq P\}$. Thus, $\bigcap\{P \in Spec^{\alpha}_{\forall}(L) \mid X \subseteq P\} \subseteq \alpha_{\forall}(X)$. Therefore, $\alpha_{\forall}(X) = \bigcap\{P \in Spec^{\alpha}_{\forall}(L) \mid X \subseteq P\}$.

(4) Take $X = \{1\}$. The result holds by (3). □

5 Conclusion

In this paper, the notion of monadic co-annihilators are firstly introduced and minimal prime monadic filters are characterized in terms of monadic co-annihilators in a monadic residuated lattice(Theorem 3.12 and Corollary 3.14). In particular, a condition is given for the lattice $\mathcal{F}^{\alpha}_{\forall}(L)$ to be a Boolean algebra (Theorem 3.16). Hyperarchimedean monadic residuated lattices are proposed and some equivalent conditions are derived for hyperarchimedean monadic residuated lattices (Theorems 3.19 and 3.20). Monadic α-filters are presented and some representations for monadic α-filters are demonstrated (Proposition 4.6 and Theorem 4.7(3)).

Acknowledgement

The author is extremely grateful to the editors and the anonymous referees for their valuable comments and constructive suggestions which help to greatly improve the presentation of this article.

References

[1] I. Calomino, Quasicomplemented distributive nearlattices, J. Mult.-Val. Log. Soft Comput. 43(2024) 167-184.

[2] I. Calomino, S. Celani, α-filters and α-order-ideals in distributive quasicomplemented semilattices, Commentat. Math. Univ. Carol. 62(2021) 15-32.

[3] D. Castaño, C. Cimadamore, J. P. Díaz Varela, L. Rueda, Monadic BL-algebras: the equivalent algebraic semantics of Hájek's monadic fuzzy logic, Fuzzy Sets Syst. 320(2017) 40-59.

[4] S. A. Celani, α-Ideals and α-deductive systems in bounded Hilbert algebras, J. of Mult.-Valued Logic and Soft Computing 21(5-6)(2013) 493-510.

[5] W. H. Cornish, Annulets and α-ideals in distributive lattice, J. Aust. Math. Soc. 15(1973) 70-77.

[6] W. H. Cornish, Congruences on distributive pseudocomplemented lattices, Bulletin of the Australian Mathematical Society 8(2)(1973) 161-179.

[7] W. H. Cornish, A sheaf representation of distributive pseudocomplemented lattices, Proceedings of the American Mathematical Society 57(1976) 11-15.

[8] A. Di Nola, R. Grigolia, On monadic MV-algebras, Ann. Pure Appl. Log. 128(2004) 125-139.

[9] Y. Y. Dong, X. L. Xin, α-filters and prime α-filters in residuated lattices, Soft Computing 23(2019) 3207-3216.

[10] L. J. González, I. Calomino, Finite distributive nearlattices, Discrete Mathematics 344(9)(2021) 112511.

[11] R. P. Halmos, Algebraic logic, I. Monadic boolean algebras, Composition Mathematica 12(1954-1956) 217-249.

[12] M. Haveshki, M. Mohamadhasani, On α-filtetrs of BL-algebras, J. Intell. and Fuzzy Syst. 28(2015) 373-382.

[13] C. Jayaram, Prime α-ideals in a 0-distributive lattice, Indian J. Pure Appl. Math. 3(1986) 331-337.

[14] J. E. Kist, Minimal prime ideals in commutative semigroups, Proc. London Math. Soc. 13(3)(1963) 31-50.

[15] L. Z. Liu, X. Zhang, The Belluce-semilattice associated with a monadic residuated lattice, Soft Comput 27(2023) 6983-6998.

[16] Y. S. Pawar D. N. Mane, α-ideals in 0-distributive semilattices and 0-distributive lattices, Indian J. Pure Appl. Math. 24(7-8)(1993) 435-443.

[17] Y. S. Pawar, S. S. Khopade, α-ideals and annihilator ideals in 0-distributive lattices, Acta. Univ. Palack. Olomuc. Fac. Med. 49(1)(2010) 63-74.

[18] J. Rachůnek, F. Švrček, Monadic bounded commutative residuated l-monoids. Order 25(2)(2008) 157-175.

[19] J. Rachůnek, D. Šalounová, Monadic bounded residuated lattices, Order 30(2013) 195-210.

[20] S. Rasouli, Generalized co-annihilators in residuated lattices, Annals of the University of Craiova 45(2)(2018) 190-207.

[21] S. Rasouli, Quasicomplemented residuated lattices, Soft Computing 24(9)(2020) 6591-6602.

[22] S. Rasouli, A. Dehghani, The hull-kernel topology on prime filters in residuated lattices, Soft Computing 25(16)(2021) 10519-10541.

[23] S. Rasouli, A. Dehghani, The pure spectrum of a residuated lattice, Fuzzy Sets and Systems 469(2023) 108-636.

[24] K. I. Rosenthal, Quantales and their applications, Pitman Res. Notes Math. Ser., Longman, 1990.

[25] T. P. Speed, Spaces of ideals of distributive lattices II. Minimal prime ideals, J. Aust. Math. Soc. 18(1974) 54-72.

[26] J. Wang, X. Xin, P. He, Monadic bounded hoops, Soft Comput. 22(2018) 1749-1762.

[27] J. Wang, P. He, Y. She, Monadic NM-algebras, Logic Journal of the IGPL 27(6)(2019) 812-835.

[28] J. Wang, M. Wang, The lattices of monadic filters in monadic BL-algebras, IAENG International Journal of Applied Mathematics 50(3)(2020) 656-660.

[29] J. T. Wang, P. F. He, J. Yang, M. Wang, X.L. He, Monadic NM-algebras: an algebraic approach to monadic predicate nilpotent minimum logic, J. Log. Comput. 32(2022) 741-766.

[30] J. T. Wang, Y. H. She, P. F. He, N.N. Ma, On categorical equivalence of weak monadic residuated distributive lattices and weak monadic c-differential residuated distributive lattices, Stud. Log. 111 (2023) 361-390.

[31] J. T. Wang, H. W. Wu, P. F. He, Y. H. She, An algebraic proof of completeness for monadic fuzzy predicate logic **mMTL**∀, The Review of Symbolic Logic 18(2025) 213-239.

Received 29 April 2025

Monadic Pseudocomplemented Distributive Lattices

Ismael Calomino
Facultad de Ciencias y Artes, Universidad Católica de Ávila, C/Canteros s/n, 05005 Ávila, España.
ismaelm.calomino@ucavila.es

Andrew Lewis-Smith
Department of Computer Science, Middlesex University, London, UK.
a.lewis-smith@mdx.ac.uk

Gustavo Pelaitay
CONICET and Instituto de Ciencias Básicas, Universidad Nacional de San Juan, 5400 San Juan, Argentina.
gpelaitay@gmail.com

Abstract

In this paper, we study the variety of pseudocomplemented distributive lattices with existential and universal quantifiers, called monadic pseudocomplemented distributive lattices. We introduce the variety of monadic KAN-algebras, which turns out to be different from the class studied in [Gomez C., Marcos M., San Martín H.J.: *On the relation of negations in Nelson algebras*. Rep. Math. Logic **56** (2021), 15–56], and prove that the category of monadic pseudocomplemented distributive lattices is equivalent to the category of centered monadic KAN-algebras, extending the results given in [Calomino I., Pelaitay G.: *A new categorical equivalence for Stone algebras*. Accepted in Mathematica Slovaca (2025)].

1 Introduction

In [17], Halmos introduces the class of monadic Boolean algebras as an algebraic counterpart of the one-variable fragment of classical predicate logic. A monadic Boolean algebra is a structure $\langle B, \exists \rangle$, where B is a Boolean algebra and $\exists$ is a unary operator on B, called the existential quantifier, satisfying the following conditions:

(MB1) $\exists 0 = 0$,

(MB2) $x \leq \exists x$,

(MB3) $\exists(x \wedge \exists y) = \exists x \wedge \exists y$,

(MB4) $\exists(x \vee y) = \exists x \vee \exists y$.

We can define the dual operator $\forall$, called the universal quantifier, as $\forall x = (\exists x')'$, where x' denotes the Boolean negation of x. It follows that $\forall$ satisfies the following dual properties:

(MB5) $\forall 1 = 1$,

(MB6) $\forall x \leq x$,

(MB7) $\forall(x \vee \forall y) = \forall x \vee \forall y$,

(MB8) $\forall(x \wedge y) = \forall x \wedge \forall y$.

Monadic operators have attracted considerable attention and have been systematically studied across a wide range of algebraic structures. Notable examples include monadic Heyting algebras [3], monadic MV-algebras [9], monadic NM-algebras [27, 28], monadic semi-Nelson algebras [15], monadic residuated distributive lattices [29, 30], monadic quasi-modal distributive nearlattices [4], monadic $k \times j$-rough Heyting algebras [1], and monadic distributive lattices [11, 12, 13]. In addition, quantifiers on distributive lattices have been investigated in [6]. Recently, in [10], an interesting connection has been established between the class of monadic Boolean algebras with an automorphism and the class $\mathbf{Df_2}$ of diagonal-free two-dimensional cylindric algebras. In a related direction, Figallo Orellano [14] has studied MV-algebras endowed with two commuting quantifiers, providing a further generalization of these algebraic frameworks.

In the monadic Boolean setting, the existential quantifier is defined in terms of the universal quantifier using Boolean negation, which plays a fundamental role in classical logic. Therefore, it is natural to study monadic operators on bounded distributive lattices, where Boolean negation is replaced by the 'logically weaker' (more general) operation of pseudocomplementation. A well-known generalization of Boolean algebras is the variety of distributive pseudocomplemented lattices, also known as distributive p-algebras. These structures have been extensively studied by several authors in various papers. Among the key works addressing their properties and structures, we mention [2, 8, 18, 19, 21, 23].

Following the results of [20, 6], Gómez et al. [16] introduced the notion of an existential quantifier for the class of pseudocomplemented distributive lattices and established a categorical equivalence between the class of pseudocomplemented distributive lattices equipped with an existential quantifier and the variety of Kleene algebras with intuitionistic negation (called KAN-algebras) equipped with a weak quantifier.

The aim of this paper is to present an alternative approach to the results obtained in [16] by studying the class of pseudocomplemented distributive lattices equipped with existential and universal quantifiers, each of which are not necessarily interdefinable via the pseudocomplement operation. We then introduce a class of monadic KAN-algebras, which differs from that considered in [16], and present several constructions based on Kalman's functor and the results given in [5].

The structure of the paper is as follows. In Section 2, we review basic notions on pseudocomplemented distributive lattices and KAN-algebras that are necessary for the subsequent developments. Section 3 introduces the class of monadic pseudocomplemented distributive lattices, establishes their fundamental properties, and presents a Glivenko-type theorem in this context. In Section 4, we define the class of monadic KAN-algebras, prove several properties, and show how they relate to monadic pseudocomplemented distributive lattices via appropriate constructions. Section 5 is devoted to various constructions for monadic KAN-algebras, including extensions of Kalman's and Monteiro's constructions to our monadic setting, as well as a generalization of Sendlewski's construction. Finally, we establish a categorical equivalence between the category of monadic pseudocomplemented distributive lattices and the category of centered monadic KAN-algebras.

2 Preliminaries

We recall some basic notions on pseudocomplemented distributive lattices and KAN-algebras that will be used throughout the rest of the paper.

2.1 Pseudocomplemented distributive lattices

A pseudocomplemented distributive lattice is an algebra $\langle A, \vee, \wedge, ^*, 0, 1\rangle$ of type $(2, 2, 1, 0, 0)$ where $\langle A, \vee, \wedge, 0, 1\rangle$ is a bounded distributive lattice, and for all $a, b \in A$, the following condition holds: $a \wedge b = 0$ if and only if $a \leq b^*$. Thus, a^* is the greatest element of A that is disjoint from a; that is, $a^* = \max\{x \in A : a \wedge x = 0\}$. We denote the class of pseudocomplemented distributive lattices by **PDL**.

Theorem 2.1. *Let $A \in$ **PDL**. Then:*

1. $0^* = 1$ *and* $1^* = 0$,

2. $x \leq x^{**}$,

3. $x^{***} = x^*$,

4. $x \leq y$ *implies* $y^* \leq x^*$,

5. $(x^{**} \wedge y^{**})^* = (x \wedge y)^*$,

6. $(x \wedge y)^{**} = x^{**} \wedge y^{**}$,

7. $(x \vee y)^* = x^* \wedge y^*$,

8. $(x^{**} \vee y^{**})^{**} = (x \vee y)^{**}$.

An element a of a pseudocomplemented distributive lattice A is called dense if $a^* = 0$, and the set $D(A)$ of all dense elements is a filter of A. An element a of A is called regular if $a^{**} = a$. Denote by $R(A)$ the set of all regular elements of A. We define a binary operation on $R(A)$ given by $x \sqcup y = (x \vee y)^{**}$.

Theorem 2.2. [2, 7] *Let $A \in$ **PDL**. Then the structure $\langle R(A), \sqcup, \wedge, ^*, 0, 1\rangle$ is a Boolean algebra and the binary relation θ_G on A defined by*

$$(x, y) \in \theta_G \iff x^* = y^* \tag{1}$$

is a congruence on A. Moreover, $R(A)$ is isomorphic to A/θ_G.

Definition 2.1. [22, 16] *Let $A \in$ **PDL**.*

- *A filter F of A is called Boolean if it contains all dense elements, or equivalently, $x \vee x^* \in F$, for all $x \in A$.*

- *A congruence θ of A is called Boolean if A/θ is a Boolean algebra, or equivalently, $x \vee x^* \in [1]_\theta$, for all $x \in A$.*

Lemma 2.1. [22, 16] *Let $A \in$ **PDL**. Then:*

1. *If θ is a Boolean congruence of A, then $[1]_\theta$ is a Boolean filter of A.*

2. *If F is a Boolean filter of A, then*

$$\theta(F) = \{(x, y) \in A \times A \colon \exists f \in F(x \wedge f = y \wedge f)\}$$

is a Boolean congruence of A.

The assignments $\theta \mapsto [1]_\theta$ and $F \mapsto \theta(F)$ define an order isomorphism between the poset of Boolean congruences of A and the poset of Boolean filters of A.

2.2 KAN-algebras

Definition 2.2. *[16, 20] An algebra* $\langle T, \vee, \wedge, \sim, c, 0, 1\rangle$ *of type* $(2, 2, 1, 0, 0, 0)$ *is called a centered Kleene algebra if the reduct* $\langle T, \vee, \wedge, 0, 1\rangle$ *is a bounded distributive lattice and* $\sim$ *satisfies the following conditions:*

1. $\sim (x \wedge y) =\sim x \vee \sim y$,
2. $\sim\sim x = x$,
3. $x \wedge \sim x \leq y \vee \sim y$,
4. $\sim c = c$.

Definition 2.3. *[16, 25] A Kleene algebra with intuitionistic negation, or KAN-algebra for short, is an algebra* $\langle T, \vee, \wedge, \sim, \neg, 0, 1\rangle$ *of type* $(2, 2, 1, 1, 0, 0)$ *such that* $\langle T, \vee, \wedge, \sim, 0, 1\rangle$ *is a Kleene algebra and the following conditions are satisfied:*

(K1) $\neg(x \wedge \neg(x \wedge y)) = \neg(x \wedge \neg y)$,

(K2) $\neg(x \vee y) = \neg x \wedge \neg y$,

(K3) $x \wedge \sim x = x \wedge \neg x$,

(K4) $\sim x \leq \neg x$,

(K5) $\neg(x \wedge y) = \neg((\sim \neg x) \wedge y)$.

An algebra $\langle T, \vee, \wedge, \sim, \neg, c, 0, 1\rangle$ *of type* $(2, 2, 1, 1, 0, 0, 0)$ *is a centered KAN-algebra, or KANc-algebra for short, if* $\langle T, \vee, \wedge, \sim, \neg, 0, 1\rangle$ *is a KAN-algebra and the structure* $\langle T, \vee, \wedge, \sim, c, 0, 1\rangle$ *is a centered Kleene algebra.*

We denote the variety of KAN-algebras by **KAN** and the variety of centered KAN-algebras by **KANc**. In any $T \in \mathbf{KAN}$ it is verified that $\neg 1 = 0$, $\neg 0 = 1$, $\neg(x \wedge \sim x) = \neg(x \wedge \neg x) = 1$, and $x \leq y$ implies $\neg y \leq \neg x$ (for more details see [16]). We adopt the following notation:

$$\Diamond x :=\sim \neg x \tag{2}$$

and

$$\Box x := \neg \sim x. \tag{3}$$

Then $\Box x =\sim \Diamond \sim x$ and $\Diamond x =\sim \Box \sim x$. The operators $\Diamond$ and $\Box$ we can describe several important properties of KAN-algebras that will be used in this paper.

Proposition 2.3. *[16, 5] Let $T \in$ **KAN**. Then:*

1. $\Diamond 0 = 0$ *and* $\Box 1 = 1$,
2. $\Diamond 1 = 1$ *and* $\Box 0 = 0$,
3. $\Diamond x \leq x \leq \Box x$,
4. $\Diamond x \leq \neg\neg x \leq \Box x$,
5. $\neg \Diamond x = \Box \neg x = \neg x$,
6. $\Diamond \Diamond x = \Diamond x$,
7. $\Diamond(x \vee y) = \Diamond x \vee \Diamond y$,
8. $\Diamond(\Diamond x \wedge \Diamond y) = \Diamond(x \wedge y)$,
9. $\Box\Box x = \Box x$,
10. $\Box(x \wedge y) = \Box x \wedge \Box y$,
11. $\Box(\Box x \vee \Box y) = \Box(x \vee y)$,
12. $x \vee \sim x = \sim x \vee \Diamond x$,
13. $x \wedge \sim x = x \wedge \Box \sim x$,
14. $x = (\Diamond x \vee \sim x) \wedge \Box x$,
15. $\Box x = \Box y$ *and* $\Diamond x = \Diamond y$ *implies* $x = y$,
16. $\Diamond(x \wedge y) = 0$ *if and only if* $\Diamond x \leq \Diamond \neg y$.

Lemma 2.2. *[16, 5] Let $T \in$ **KANc**. Then:*

1. $\neg c = 1$,
2. $\Diamond c = 0$ *and* $\Box c = 1$,
3. $\Diamond(c \wedge \Box x) = 0$ *and* $\Box(\Diamond x \vee c) = 1$,
4. $x = (\Diamond x \vee c) \wedge \Box x$,
5. $x \vee c = \Diamond x \vee c$.

Let $T \in \mathbf{KAN}$ and consider the set $T^{\Diamond} := \{x \in T \colon \Diamond x = x\}$. We define on $T^{\Diamond}$ the operations $x \vee^{\Diamond} y := \Diamond(x \vee y)$, $x \wedge^{\Diamond} y := \Diamond(x \wedge y)$ and $\neg^{\Diamond} x := \Diamond \neg x$. It is easy to see that $x \vee^{\Diamond} y = x \vee y$, and if $x \in T^{\Diamond}$, then $\sim x = \neg x$. Hence, we have $T^{\Diamond} = \{x \in T \colon \sim x = \neg x\}$. If we consider the set $T^{\Box} := \{x \in T \colon \Box x = x\}$, then $T^{\Box} = \sim (T^{\Diamond})$, and by the results given in [16, 5], the structure $\langle T^{\Diamond}, \vee, \wedge^{\Diamond}, \neg^{\Diamond}, 0, 1\rangle$ is a pseudocomplemented distributive lattice.

On the other hand, in [16], the authors proved that if T is a KAN-algebra, then the binary relation $\theta \subseteq T \times T$ defined by $(x, y) \in \theta$ if and only if $\neg x = \neg y$ is an equivalence relation on T that is compatible with the operations $\vee$, $\wedge$, and $\neg$. If we denote by $[x]_\theta$ the equivalence class of x modulo θ and $T^\theta = \{[x]_\theta \colon x \in T\}$ the set of all equivalence classes, then $\langle T^\theta, \vee^\theta, \wedge^\theta, \neg^\theta, [0]_\theta, [1]_\theta\rangle$ is a pseudocomplemented distributive lattice, where the operations on T^θ are defined by $[x]_\theta \vee^\theta [y]_\theta := [x \vee y]_\theta$, $[x]_\theta \wedge^\theta [y]_\theta := [x \wedge y]_\theta$ and $\neg^\theta [x]_\theta := [\neg x]_\theta$. So, the order $\leq$ in T^θ is given by $[x]_\theta \leq [y]_\theta$ if and only if $\neg y \leq \neg x$. Moreover, the pseudocomplemented distributive lattices $\langle T^{\Diamond}, \vee, \wedge^{\Diamond}, \neg^{\Diamond}, 0, 1\rangle$ and $\langle T^\theta, \vee^\theta, \wedge^\theta, \neg^\theta, [0]_\theta, [1]_\theta\rangle$ are isomoprphics [5].

3 Monadic pseudocomplemented distributive lattices

In this section, we will introduce the class of monadic pseudocomplemented distributive lattices and obtain some of their properties.

Definition 3.1. *A monadic pseudocomplemented distributive lattice is a structure* $\langle A, \exists, \forall\rangle$*, where* A *is a pseudocomplemented distributive lattice and* $\exists$ *and* $\forall$ *are unary operators on* A *verifying* (MB1) *to* (MB8) *and the following identities:*

(MP9) $\forall \exists x = \exists x$,

(MP10) $\exists \forall x = \forall x$,

(MP11) $(\forall x)^* \leq \exists x^*$,

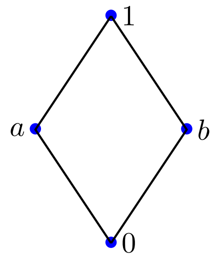
(MP12) $(\exists x)^* \leq \forall x^*$,

(MP13) $\forall(x \vee y) \leq \exists x \vee \forall y$,

(MP14) $\exists x \wedge \forall y \leq \exists(x \wedge y)$.

Denote the variety of monadic pseudocomplemented distributive lattices by **MPDL**.

Example 3.1. *Consider the four-element diamond $A = \{0, a, b, 1\}$, whose Hasse diagram and pseudocomplementation table are as follows:*



x	x^*
0	1
a	b
b	a
1	0

The monadic operators $\exists$ and $\forall$ are defined by:

x	$\exists x$	$\forall x$
0	0	0
a	1	0
b	1	0
1	1	1

Thus, $\langle A, \exists, \forall \rangle$ is a monadic pseudocomplemented distributive lattice.

In the following lemma we obtain some properties of monadic operators.

Lemma 3.1. *Let $\langle A, \exists, \forall \rangle \in$ **MPDL**. Then:*

1. $\exists 1 = 1$ *and* $\forall 0 = 0$,
2. $x \leq y$ *implies* $\forall x \leq \forall y$ *and* $\exists x \leq \exists y$,
3. $\exists\exists x = \exists x$ *and* $\forall\forall x = \forall x$,
4. $x \leq \exists y$ *if and only if* $\exists x \leq \exists y$,
5. $\forall x \leq y$ *if and only if* $\forall x \leq \forall y$,
6. $\exists x = x$ *if and only if* $\forall x = x$,
7. $\exists x \leq y$ *if and only if* $x \leq \forall y$,
8. $\forall x = 1$ *if and only if* $x = 1$,
9. $\exists x = 0$ *if and only if* $x = 0$.

Proof. It is routine. □

Remark 3.1. *Let P and Q be two posets. A pair (f, g) of maps $f\colon P \to Q$ and $g\colon Q \to P$ forms an adjunction (or Galois connection) between P and Q if, for every $p \in P$ and $q \in Q$, we have $f(p) \leq q$ if and only if $p \leq g(q)$. By item (7) of Lemma 3.1, if $A \in$ **MPDL**, then the pair $(\exists, \forall)$ forms a Galois connection on A.*

The following lemma illustrates the relationship between the monadic operators and pseudocomplementation in a pseudocomplemented distributive lattice.

Lemma 3.2. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL**. Then:*

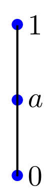
1. $\forall x^* = (\exists x)^*$,
2. $\exists x^* = (\forall x)^*$,
3. $(\forall x)^{**} = \forall x^{**}$,
4. $(\exists x)^{**} = \exists x^{**}$,
5. $\exists x \leq (\forall x^*)^*$,
6. $\forall x \leq (\exists x^*)^*$,
7. $\forall(\exists x \vee y) = \exists x \vee \forall y$,
8. $\exists(\forall x \wedge y) = \forall x \wedge \exists y$.

Proof. We will prove only (1), (5) and (7).
(1): By (MP14) and (MB1) we have $\exists x \wedge \forall x^* \leq \exists(x \wedge x^*) = \exists 0 = 0$. So, $\forall x^* \wedge \exists x = 0$ and $\forall x^* \leq (\exists x)^*$. Then, by (MP12) we have $\forall x^* = (\exists x)^*$.
(5): By (MP14) and (MB1), it follows $\exists x \wedge \forall x^* \leq \exists(x \wedge x^*) = 0$, i.e., $\exists x \leq (\forall x^*)^*$.
(7): By (MP9), (MB7) and Lemma 3.1 we have

$$\forall(\exists x \vee y) = \forall(\forall\exists x \vee y) = \forall\forall\exists x \vee \forall y = \exists x \vee \forall y.$$

Therefore, (7) is proved. □

Remark 3.2. *Consider the three-element chain $A = \{0, a, 1\}$, whose Hasse diagram and pseudocomplementation table are as follows:*



x	x^*
0	1
a	0
1	0

The monadic operators $\exists$ *and* $\forall$ *are defined by:*

x	$\exists x$	$\forall x$
0	0	0
a	1	a
1	1	1

Thus, $\langle A, \exists, \forall \rangle$ *is a monadic pseudocomplemented distributive lattice, and*

$$(\exists a^*)^* = 1 \not\leq a = \forall a.$$

Therefore, unlike in the Boolean case, in the class of monadic pseudocomplemented distributive lattices the monadic operators $\exists$ *and* $\forall$ *are not interdefinable by means of pseudocomplementation.*

Lemma 3.3. *Let* $\langle A, \exists, \forall \rangle \in \mathbf{MPDL}$ *and define* $M := \{x \in A \colon \exists x = x\}$*. Then:*

1. $M = \{x \in A \colon \forall x = x\}$.

2. M *is a subalgebra of* $\langle A, \exists, \forall \rangle$.

3. $\exists x = \min\{y \in M \colon x \leq y\}$ *and* $\forall x = \max\{y \in M \colon y \leq x\}$.

Proof. (1): This follows from Lemma 3.1.
(2): By (MB4), (MB8), (MP9), and (MP10), the set M is closed under $\vee$, $\wedge$, $\forall$, and $\exists$, respectively. Since $\exists 0 = 0$ and $\exists 1 = 1$, it follows that $0, 1 \in M$. Finally, by Lemma 3.1, if $x \in M$, then $x^* = (\exists x)^* = \forall x^*$ and $x^* = (\forall x)^* = \exists x^*$. Thus, M is closed under * and hence M is a subalgebra of $\langle A, \forall, \exists \rangle$.
(3): We only prove $\exists x = \min\{y \in M \colon x \leq y\}$. Since $\exists x = \forall \exists x$ and $x \leq \exists x$, then $\exists x \in M$ and $\min\{y \in M \colon x \leq y\} \leq \exists x$. On the other hand, if $y \in M$ such that $x \leq y$, then $\exists x \leq \exists y = y$ and $\exists x \leq \min\{y \in M \colon x \leq y\}$. □

3.1 Glivenko's theorem

We prove a Glivenko style theorem for monadic pseudocomplemented distributive lattices, as a generalization of Theorem 2.2.

If $\langle A, \exists, \forall \rangle \in \mathbf{MPDL}$, then we define the operator

$$\exists_r x := (\exists x)^{**}. \tag{4}$$

By Lemma 3.2 we have $\exists_r x = \exists x^{**}$. The following result extends the Theorem 2.2.

Theorem 3.1. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL**. Then $\langle R(A), \exists_r\rangle$ is a monadic Boolean algebra and the binary relation θ_G given by 1 is a monadic Boolean congruence on A. Moreover, the monadic Boolean algebras $R(A)$ and A/θ_G are isomorphic.*

Proof. It is clear that $\exists_r$ is well defined over $R(A)$. Since $\exists_r x = \exists x$ for all $x \in R(A)$, then it is easy to prove (MB1), (MB2) and (MB3). We see (MB4). Let $x, y \in R(A)$. Then by Theorem 2.1 and Lemma 3.2,

$$\begin{aligned} \exists_r(x \sqcup y) &= (\exists(x \vee y)^{**})^{**} &&= (\exists(x \vee y))^{**} \\ &= (\exists x \vee \exists y)^{**} &&= ((\exists x)^{**} \vee (\exists y)^{**})^{**} \\ &= (\exists_r x \vee \exists_r y)^{**} &&= \exists_r x \sqcup \exists_r y. \end{aligned}$$

Therefore, the structure $\langle R(A), \exists_r\rangle$ is a monadic Boolean algebra. Now, we prove that θ_G is compatible with the operators $\exists$ and $\forall$. If $(x, y) \in \theta_G$, then $x^* = y^*$. So, by Lemma 3.2, it follows $(\exists x)^* = \forall x^* = \forall y^* = (\exists y)^*$ and $(\exists x, \exists y) \in \theta_G$. Similarly, $(\forall x, \forall y) \in \theta_G$ and θ_G is a congruence on A. Finally, it follows that the monadic Boolean algebras $R(A)$ and A/θ_G are isomorphic by the homomorphism $\varphi\colon A \to R(A)$ given by $\varphi(x) = x^{**}$. □

3.2 Monadic Boolean filters and monadic Boolean congruences

We introduce the notions of monadic Boolean filters and monadic Boolean congruences, and establish a correspondence between them in the context of monadic pseudocomplemented distributive lattices.

Definition 3.2. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL**.*

- *A Boolean filter F of A is called monadic if it satisfies the following condition: $x \in F$ implies $\forall x \in F$.*
- *A Boolean congruence θ of A is called monadic if A/θ is a monadic Boolean algebra.*

Remark 3.3. *Note that if F is a monadic Boolean filter of a monadic pseudocomplemented distributive lattice $\langle A, \exists, \forall\rangle$ and $x \in F$, then $\exists x \in F$.*

Remark 3.4. *Consider Example 3.1. Then the sets $F_a = \{a, 1\}$ and $F_b = \{b, 1\}$ are Boolean filters that are closed under $\exists$, but not under $\forall$.*

Lemma 3.4. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL**. Then:*

1. *If θ is a monadic Boolean congruence of $\langle A, \exists, \forall\rangle$, then $[1]_\theta$ is a monadic Boolean filter of $\langle A, \exists, \forall\rangle$.*

2. *If F is a monadic Boolean filter of $\langle A, \exists, \forall\rangle$, then $\theta(F)$ is a monadic Boolean congruence of $\langle A, \exists, \forall\rangle$.*

Moreover, the assignments $\theta \mapsto [1]_\theta$ and $F \mapsto \theta(F)$ define an order isomorphism between the poset of monadic Boolean congruences of $\langle A, \exists, \forall\rangle$ and the poset of monadic Boolean filters of $\langle A, \exists, \forall\rangle$.

Proof. (1) It is an immediate consequence of (MB5).
(2) Let F be a monadic Boolean filter of $\langle A, \exists, \forall\rangle$. Then $\theta(F)$ is a Boolean congruence. We prove that $\theta(F)$ is compatible with $\forall$. If $(x, y) \in \theta(F)$, then there exists $f \in F$ such that $x \wedge f = y \wedge f$. So, by (MB8), $\forall x \wedge \forall f = \forall y \wedge \forall f$ and since F is monadic filter, $\forall f \in F$. Then $(\forall x, \forall y) \in \theta(F)$.

Now, we prove that $\theta(F)$ is compatible with $\exists$. If $(x, y) \in \theta(F)$, then $(x^*, y^*) \in \theta(F)$ and by Lemma 3.2, $(\forall x^*, \forall y^*) = ((\exists x)^*, (\exists y)^*) \in \theta(F)$, i.e., there exists $f \in F$ such that $(\exists x)^* \wedge f = (\exists y)^* \wedge f$. Since F is a Boolean filter, we have $\exists x \vee (\exists x)^* \in F$ and $\exists y \vee (\exists y)^* \in F$. We take $g = (\exists x \vee (\exists x)^*) \wedge (\exists y \vee (\exists y)^*) \wedge f \in F$. Then

$$\begin{aligned} \exists x \wedge g &= \exists x \wedge (\exists x \vee (\exists x)^*) \wedge (\exists y \vee (\exists y)^*) \wedge f \\ &= \exists x \wedge (\exists y \vee (\exists y)^*) \wedge f \\ &= (\exists x \wedge \exists y \wedge f) \vee (\exists x \wedge (\exists y)^* \wedge f) \\ &= (\exists x \wedge \exists y \wedge f) \vee (\exists x \wedge (\exists x)^* \wedge f) \\ &= \exists x \wedge \exists y \wedge f. \end{aligned}$$

Analogously, we can prove $\exists y \wedge g = \exists x \wedge \exists y \wedge f$. Therefore, $(\exists x, \exists y) \in \theta(F)$ and $\theta(F)$ is a monadic Boolean congruence. □

Remark 3.5. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL**. Since $D(A)$ is a monadic Boolean filter, it follows by Lemma 3.4 that $\theta(D(A))$ is a monadic Boolean congruence. Moreover, it can be shown that this congruence coincides with the Glivenko congruence θ_G defined in 1.*

4 Monadic KAN-algebras

The aim of this section is to introduce the class of monadic KAN-algebras and establish some of their properties.

For any $T \in$ **KAN** and unary operator $\exists\colon T \to T$ we define the operator

$$\forall x := \sim \exists \sim x. \tag{5}$$

It is easy to verify that $\exists x = \sim \forall \sim x$.

Definition 4.1. *A monadic KAN-algebra is a structure $\langle T, \exists \rangle$, where T is a KAN-algebra and $\exists$ is a unary operator on T verifying the following identities:*

(MK1) $\exists 0 = 0$,

(MK2) $x \leq \exists x$,

(MK3) $\exists(x \wedge \exists y) = \exists x \wedge \exists y$,

(MK4) $\exists(x \vee y) = \exists x \vee \exists y$,

(MK5) $\forall \exists x = \exists x$,

(MK6) $\forall x \wedge \exists y \leq \exists(x \wedge y)$,

(MK7) $\forall \neg x = \neg \exists x$,

(MK8) $\neg \forall x = \exists \neg x$.

Furthermore, if T is a KAN-algebra with center c and $\exists c = c$, we refer to $\langle T, \exists, c \rangle$ as a centered monadic KAN-algebra.

We denote the variety of monadic KAN-algebras by **MKAN** and the variety of centered monadic KAN-algebras by **MKANc**.

Lemma 4.1. *Let $\langle T, \exists \rangle \in$ **MKAN**. Then:*

1. $\exists 1 = 1$,
2. $\exists \exists x = \exists x$,
3. $\forall 1 = 1$ *and* $\forall 0 = 0$,
4. $\forall x \leq x$ *and* $\forall \forall x = \forall x$,
5. $\forall(x \vee \forall y) = \forall x \vee \forall y$,
6. $\forall(x \wedge y) = \forall x \wedge \forall y$,
7. $\exists \forall x = \forall x$,
8. $\forall \sim x \leq \neg \exists x$,
9. $\exists \sim x \leq \neg \forall x$,
10. $\forall(x \vee y) \leq \exists x \vee \forall y$.

Proof. We only prove (5), (7) and (10).
(5): By (MK3),

$$\begin{aligned}\forall(x \vee \forall y) &= \sim\exists\sim(x\vee\sim\exists\sim y) &&= \sim\exists(\sim x\wedge\exists\sim y)\\ &= \sim\exists\sim x\vee\sim\exists\sim y &&= \forall x\vee\forall y.\end{aligned}$$

(7): By (MK5) it follows that

$$\begin{aligned}\exists\forall x &= \exists\sim\exists\sim x &&= \sim\forall\exists\sim x\\ &= \sim\exists\sim x &&= \forall x.\end{aligned}$$

(10): By (MK6) we have $\forall\sim x\wedge\exists\sim y\leq\exists(\sim x\wedge\sim y)=\exists\sim(x\vee y)$. Then, $\sim\exists\sim(x\vee y)\leq\sim(\forall\sim x\wedge\exists\sim y)=\sim\forall\sim x\vee\sim\exists\sim y$. So, $\forall(x\vee y)\leq\exists x\vee\forall y$. □

The operators $\Diamond$ and $\Box$ can be viewed as modal operators on KAN-algebras. The following lemma highlights the connection between these modal operators and the monadic operators.

Lemma 4.2. *Let* $\langle T,\exists\rangle\in\mathbf{MKAN}$. *Then:*

1. $\Diamond\exists x=\exists\Diamond x$ *and* $\Box\exists x=\exists\Box x$.

2. $\Diamond\forall x=\forall\Diamond x$ *and* $\Box\forall x=\forall\Box x$.

Proof. (1): By (2), (5) and (MK7) we have

$$\Diamond\exists x=\sim\neg\exists x=\sim\forall\neg x=\exists\sim\neg x=\exists\Diamond x.$$

Similarly, by (2), (5) and (MK8) we have

$$\Box\exists x=\neg\sim\exists x=\neg\forall\sim x=\exists\neg\sim x=\exists\Box x.$$

(2): By (2), (5) and (MK8) we have

$$\Diamond\forall x=\sim\neg\forall x=\sim\exists\neg x=\sim\exists\sim\Diamond x=\forall\Diamond x.$$

Analogously, by (3), (5) and (MK7) it follows

$$\Box\forall x=\neg\sim\forall\sim\sim x=\neg\exists\sim x=\forall\neg\sim x=\forall\Box x$$

and $\Box\forall x=\forall\Box x$, as desired. □

The following result is a consequence of Lemmas 2.2, 4.1, and 4.2.

Lemma 4.3. *Let* $\langle T,\exists,c\rangle\in\mathbf{MKANc}$. *Then:*

1. $\forall c = c$,
2. $\Diamond\exists c = 0$ *and* $\Box\exists c = 1$,
3. $\Diamond\forall c = 0$ *and* $\Box\forall c = 1$,
4. $\forall(x \vee c) = \forall x \vee c$ *and* $\forall(x \wedge c) = \forall x \wedge c$,
5. $\exists(x \vee c) = \exists x \vee c$ *and* $\exists(x \wedge c) = \exists x \wedge c$.

Let $\langle T, \exists\rangle \in \mathbf{MKAN}$. It is known that $\langle T^{\Diamond}, \vee, \wedge^{\Diamond}, \neg^{\Diamond}, 0, 1\rangle$ is a pseudocomplemented distributive lattice (see [5, 16]). Consider the following operators on $T^{\Diamond}$:

- $\exists^{\Diamond} x := \Diamond\exists x$,
- $\forall^{\Diamond} x := \Diamond\forall x$.

If $x \in T^{\Diamond}$, then $\exists^{\Diamond} x = \exists x$ and $\forall^{\Diamond} x = \forall x$. This leads to the following result.

Theorem 4.1. *Let* $\langle T, \exists\rangle \in \mathbf{MKAN}$. *Then* $\langle T^{\Diamond}, \exists^{\Diamond}, \forall^{\Diamond}\rangle$ *is a monadic pseudocomplemented distributive lattice.*

Proof. By the results developed in [16, 5] we know that $T^{\Diamond}$ is a pseudocomplemented distributive lattice. By Lemma 4.2 it follows that the operators $\exists^{\Diamond}$ and $\forall^{\Diamond}$ are well defined. Since $\Diamond\exists x = \exists x$ and $\Diamond\forall x = \forall x$ for all $x \in T^{\Diamond}$, we only prove (MB3) and (MP14). Let $x, y \in T^{\Diamond}$. Then:
(MB3): By (MB3), Proposition 2.3 and Lemma 4.2 we have

$$\exists^{\Diamond}(x \wedge^{\Diamond} \exists^{\Diamond} y) = \exists\Diamond(x \wedge \exists y) = \Diamond(\exists x \wedge \exists y) = \exists^{\Diamond} x \wedge^{\Diamond} \exists^{\Diamond} y.$$

(MP14): By (MP14), Proposition 2.3 and Lemma 4.2 it follows that

$$\exists^{\Diamond} x \wedge^{\Diamond} \forall^{\Diamond} y = \Diamond(\exists x \wedge \forall y) \leq \Diamond\exists(x \wedge y) = \exists\Diamond(\Diamond x \wedge \Diamond y) = \exists^{\Diamond}(x \wedge^{\Diamond} y).$$

The rest of the points are easily followed. □

Proposition 4.2. *Let* $\langle T_1, \exists_1\rangle, \langle T_2, \exists_2\rangle \in \mathbf{MKAN}$. *If* $f\colon T_1 \to T_2$ *is a homomorphism in* $\mathbf{MKAN}$, *then* $f^{\Diamond}\colon T_1^{\Diamond} \to T_2^{\Diamond}$ *defined by* $f^{\Diamond}(x) = \Diamond f(x)$ *is a homomorphism in* $\mathbf{MPDL}$.

Proof. We show that $f^{\Diamond}$ is compatible with $\exists$. If $x \in T^{\Diamond}$, then by Lemma 4.2 and Proposition 2.3 we have

$$\exists^{\Diamond} f^{\Diamond}(x) = \Diamond\exists\Diamond f(x) = \Diamond\exists f(x) = \Diamond f(\exists x) = f^{\Diamond}(\exists^{\Diamond} x).$$

Analogously, $\forall^{\Diamond} f^{\Diamond}(x) = f^{\Diamond}(\forall^{\Diamond} x)$ and $f^{\Diamond}$ is a homomorphism in $\mathbf{MPDL}$. □

If $\langle T, \exists\rangle \in \mathbf{MKAN}$, then the structure $\langle T^\theta, \vee^\theta, \wedge^\theta, \neg^\theta, [0]_\theta, [1]_\theta\rangle$ is a pseudocomplemented distributive lattice (see [5, 16]). We define on T^θ the following operators:

- $\exists^\theta[x]_\theta := [\exists x]_\theta$,
- $\forall^\theta[x]_\theta := [\forall x]_\theta$.

Theorem 4.3. *Let $\langle T, \exists\rangle \in \mathbf{MKAN}$. Then $\langle T^\theta, \exists^\theta, \forall^\theta\rangle$ is a monadic pseudocomplemented distributive lattice. Moreover, $\langle T^\theta, \exists^\theta, \forall^\theta\rangle$ and $\langle T^\Diamond, \exists^\Diamond, \forall^\Diamond\rangle$ are isomorphic.*

Proof. It is easy to prove that T^θ is a monadic pseudocomplemented distributive lattice. We know that the map $\varphi\colon T^\theta \to T^\Diamond$ given by $\varphi([x]_\theta) := \Diamond x$ is an isomorphism between the pseudocomplemented distributive lattices T^θ and $T^\Diamond$ [5]. We see that φ is compatible with the monadic operators. By Lemma 4.2 and Proposition 2.3 we have

$$\varphi(\exists^\theta[x]_\theta) = \varphi([\exists x]_\theta) = \Diamond\exists x = \Diamond\exists\Diamond x = \exists^\Diamond\varphi([x]_\theta).$$

In an analogous way it is proved that $\varphi(\forall^\theta[x]_\theta) = \forall^\Diamond\varphi([x]_\theta)$, as desired. □

If $f\colon T_1 \to T_2$ is a morphism between two monadic KAN-algebras $\langle T_1, \exists_1\rangle$ and $\langle T_2, \exists_2\rangle$, then the map $f^\theta\colon T_1^\theta \to T_2^\theta$ defined by $f^\theta([x]_\theta) = [f(x)]_\theta$ is a morphism between monadic pseudocomplemented distributive lattices. Therefore, θ is a functor from **MKAN** to **MPDL**.

Let $\langle T, \exists, c\rangle \in \mathbf{MKANc}$. Consider $C(T) := \{x \in T : x \geq c\}$. Then the structure $\langle C(T), \vee, \wedge, \neg^c, c, 1\rangle$, where $\neg^c x := \neg x \vee c$, is a pseudocomplemented distributive lattice (see [5]). We have the following result.

Theorem 4.4. *Let $\langle T, \exists, c\rangle \in \mathbf{MKANc}$. Then $\langle C(T), \exists, \forall\rangle$ is a monadic pseudocomplemented distributive lattice. Moreover, $\langle C(T), \exists, \forall\rangle$ and $\langle T^\Diamond, \exists^\Diamond, \forall^\Diamond\rangle$ are isomorphic.*

Proof. By Lemma 4.3 we have $\exists x = \exists x \vee c$ and $\forall x = \forall x \vee c$ for all $x \in C(T)$. Thus, it is easy to prove (MB1)-(MB8) and (MP9)-(MP14).

Let $h\colon T^\Diamond \to C(T)$ be the map defined by $h(x) := x \vee c$. We will show that h commutes with the monadic operators. If $x \in T^\Diamond$, then by Lemma 4.3, we have

$$h(\exists^\Diamond x) = \Diamond\exists x \vee c = \exists(x \vee c) = \exists h(x).$$

Analogously, $h(\forall^\Diamond x) = \forall h(x)$. So, $\langle C(T), \exists, \forall\rangle$ and $\langle T^\Diamond, \exists^\Diamond, \forall^\Diamond\rangle$ are isomorphic. □

Let $\langle T, \exists, c\rangle, \langle S, \exists, c\rangle \in \mathbf{MKANc}$. If $f\colon T \to S$ is a homomorphism of centered monadic KAN-algebras, then $C(f)\colon C(T) \to C(S)$ defined by $C(f)(x) = f(x)$ is a homomorphism of monadic pseudocomplemented distributive lattices. Therefore the assignments $T \mapsto C(T)$ and $f \mapsto C(f)$ determine a functor C from **MKANc** to **MPDL**.

5 Constructions for monadic KAN-algebras

5.1 Kalman's construction

In [26], Kalman's construction was extended to the context of Heyting algebras, leading to a categorical equivalence between Heyting algebras and centered Nelson algebras. This construction was further generalized in [16] to the setting of pseudo-complemented distributive lattices and centered KAN-algebras. In this subsection, we extend these results to the monadic setting.

Let $\langle A, \exists, \forall \rangle \in \mathbf{MPDL}$. Consider the set

$$K(A) := \{(x, y) \in A \times A \colon x \wedge y = 0\}$$

with the following operations on $K(A)$:

$$\begin{array}{rcl} (a,b) \vee (d,e) & := & (a \vee d, b \wedge e), \\ (a,b) \wedge (d,e) & := & (a \wedge d, b \vee e), \\ \sim (a,b) & := & (b,a), \\ \neg(a,b) & := & (a^*, a), \\ 0_K & := & (0,1), \\ 1_K & := & (1,0), \\ c_K & := & (0,0). \end{array}$$

By the results developed in [16] we have that $\langle K(A), \vee, \wedge, \sim, \neg, c_K, 0_K, 1_K \rangle$ is a centered KAN-algebra. So, the order on $K(A)$ is given by $(a,b) \leq (d,e)$ if and only if $a \leq d$ and $e \leq b$. We define on $K(A)$ the following operators:

$$\exists_K(a,b) = (\exists a, \forall b) \text{ and } \forall_K(a,b) = (\forall a, \exists b).$$

Then we have $\forall_K(a,b) = \sim \exists_K \sim (a,b)$. Indeed,

$$\begin{array}{rcl} \forall_K(a,b) & = & (\forall a, \exists b) \\ & = & \sim (\exists b, \forall a) \\ & = & \sim \exists_K(b,a) \\ & = & \sim \exists_K \sim (a,b). \end{array}$$

We have the following result.

Theorem 5.1. *Let $\langle A, \exists, \forall \rangle \in \mathbf{MPDL}$. Then $\langle K(A), \exists_K \rangle \in \mathbf{MKANc}$.*

Proof. Let $(a,b), (d,e) \in K(A)$. First we see that $\exists_K$ is well-defined. Since $(a,b) \in K(A)$ we have $a \wedge b = 0$. Then by (MP14) it follows that $\exists a \wedge \forall b \leq \exists(a \wedge b) = \exists 0 = 0$, i.e., $\exists_K(a,b) \in K(A)$. Analogously, we have $\forall_K(a,b) \in K(A)$.

(MK1): By (MB1) and (MB5) we have $\exists_K(0,1) = (\exists 0, \forall 1) = (0,1)$.

(MK2): Since $a \leq \exists a$ and $\forall b \leq b$ by (MB2) and (MB6), respectively, we have $(a,b) \leq (\exists a, \forall b) = \exists_K(a,b)$.

(MK3): By (MB3) and (MB7) it follows

$$\begin{array}{rcl} \exists_K((a,b) \wedge \exists_K(d,e)) & = & \exists_K(a \wedge \exists d, b \vee \forall e) \\ & = & (\exists a \wedge \exists d, \forall b \vee \forall e) \\ & = & (\exists a, \forall b) \wedge (\exists d, \forall e) \\ & = & \exists_K(a,b) \wedge \exists_K(d,e). \end{array}$$

(MK4): We have by (MB4) and (MB8) that

$$\begin{array}{rcl} \exists_K((a,b) \vee (d,e)) & = & (\exists(a \vee d), \forall(b \wedge e)) \\ & = & (\exists a \vee \exists d, \forall b \wedge \forall e) \\ & = & (\exists a, \forall b) \vee (\exists d, \forall e) \\ & = & \exists_K(a,b) \vee \exists_K(d,e). \end{array}$$

(MK5): By (MP9) and (MP10), $\forall_K \exists_K(a,b) = (\forall\exists a, \exists\forall b) = (\exists a, \forall b) = \exists_K(a,b)$.

(MK6): By (MP13) and (MP14) it follows

$$\begin{array}{rcl} \forall_K(a,b) \wedge \exists_K(d,e) & = & (\forall a \wedge \exists d, \exists b \vee \forall e) \\ & \leq & (\exists(a \wedge d), \forall(b \vee e)) \\ & = & \exists_K(a \wedge d, b \vee e) \\ & = & \exists_K((a,b) \wedge (d,e)). \end{array}$$

(MK7): By Lemma 3.2 (1) we have

$$\begin{array}{rcl} \forall_K \neg(a,b) & = & (\forall a^*, \exists a) \\ & = & ((\exists a)^*, \exists a) \\ & = & \neg(\exists a, \forall b) \\ & = & \neg\exists_K(a,b). \end{array}$$

(MK8): By Lemma 3.2 (2) it follows that

$$\begin{array}{rcl} \neg\forall_K(a,b) & = & ((\forall a)^*, \forall a) \\ & = & (\exists a^*, \forall a) \\ & = & \exists_K(a^*, a) \\ & = & \exists_K \neg(a,b). \end{array}$$

Finally, by (MB1) and Lemma 3.1 (1), we have $\exists_K(0,0) = (\exists 0, \forall 0) = (0,0)$ and the structure $\langle K(A), \exists_K \rangle$ is a centered monadic KAN-algebra. ☐

Remark 5.1. *Let $\langle A, \exists, \forall \rangle \in$ **MPDL**. Since $\langle K(A), \exists_K \rangle$ is a centered monadic KAN-algebra, then we note that if $(a,b) \in K(A)$ we have*

$$\Diamond(a,b) = \sim \neg(a,b) = \sim (a^*, a) = (a, a^*)$$

and

$$\Box(a,b) = \neg \sim (a,b) = \neg(b,a) = (b^*, b).$$

Then $K(A)^{\Diamond} = \{(a,a^*) : a \in A\}$ and $K(A)^{\Box} = \{(b^*, b) : b \in A\}$.

We denote by $\mathbb{MPDL}$ the category whose objects are monadic pseudocomplemented distributive lattices and by $\mathbb{MKANC}$ the category whose objects are centered monadic KAN-algebras. In both cases, the morphisms are the corresponding algebra homomorphisms. Furthermore, if $\langle A, \exists, \forall \rangle$ and $\langle B, \exists, \forall \rangle$ are two monadic pseudocomplemented distributive lattices and $h\colon A \to B$ is a morphism in $\mathbb{MPDL}$, then it is easy to see that the map $K(h)\colon K(A) \to K(B)$ defined by $K(h)(x,y) = (h(x), h(y))$ is a morphism in $\mathbb{MKANC}$. It is evident that these assignments establish a functor K from $\mathbb{MPDL}$ to $\mathbb{MKANC}$. Moreover, if $\langle T, \exists \rangle$ and $\langle S, \exists \rangle$ are two centered monadic KAN-algebras and $f\colon T \to S$ is a homomorphism of monadic centered KAN-algebras, then $C(f)\colon C(T) \to C(S)$ defined by $C(f)(x) = f(x)$ is a homomorphism of monadic pseudocomplemented distributive lattices. Is clear that the assignments $T \mapsto C(T)$ and $f \mapsto C(f)$ determine a functor C from $\mathbb{MKANC}$ to $\mathbb{MPDL}$.

Lemma 5.1. *Let $\langle A, \exists, \forall \rangle \in$ **MPDL**. Then the map $\alpha\colon A \to C(K(A))$ defined by $\alpha(x) := (x, 0)$ is an isomorphism in $\mathbb{MPDL}$.*

Proof. We only prove that α is compatible with the existential and universal quantifiers. If $x \in A$, then by Lemma 3.1 (1) and (MB1) we have

$$\alpha(\exists x) = (\exists x, 0) = (\exists x, \forall 0) = \exists_K(x,0) = \exists_K \alpha(x)$$

and

$$\alpha(\forall x) = (\forall x, 0) = (\forall x, \exists 0) = \forall_K(x,0) = \forall_K \alpha(x)$$

as claimed. ☐

Lemma 5.2. *Let $\langle T, \exists, c \rangle \in$ **MKANc**. Then the map $\beta\colon T \to K(C(T))$ defined by $\beta(x) := (x \vee c, \sim x \vee c)$ is an isomorphism in $\mathbb{MKANC}$.*

Proof. We only prove that β commutes with the existential quantifier. Let $x \in T$. Then by Lemma 4.3,

$$\begin{aligned} \beta(\exists x) &= (\exists x \vee c, \sim \exists x \vee c) \\ &= (\exists x \vee c, \forall \sim x \vee c) \\ &= (\exists(x \vee c), \forall(\sim x \vee c)) \\ &= \exists_K(x \vee c, \sim x \vee c) \\ &= \exists_K \beta(x). \end{aligned}$$

Therefore, $\langle T, \exists \rangle$ and $\langle K(C(T)), \exists_K \rangle$ are isomorphic. □

Remark 5.2. *The isomorphism β from Lemma 5.2 also commutes with the universal quantifier $\forall$. This follows from the fact that β commutes with $\sim$ and with $\exists$, together with equation* (5)*.*

By the results established in this subsection, we prove the following theorem.

Theorem 5.2. *The functors K and C establish a categorical equivalence between $\mathbb{MPDL}$ and $\mathbb{MKANC}$ with natural isomorphism α and β.*

We conclude this subsection by presenting an example that highlights the motivation behind studying monadic pseudocomplemented distributive lattices.

Let $T \in \mathbf{KAN}$. In [16], the authors introduced the notion of a weak quantifier as a map $\Omega\colon T \to T$ satisfying the following conditions:

(W1) $\Omega 0 = 0$,

(W2) $\neg\Omega x \leq \neg x$,

(W3) $\neg\Omega(x \wedge \Omega y) = \neg(\Omega x \wedge \Omega y)$,

(W4) $\Omega(x \vee y) = \Omega x \vee \Omega y$,

(W5) $\neg\Omega x = \sim \Omega x$,

(W6) $\Omega x = \Omega \sim \neg x$.

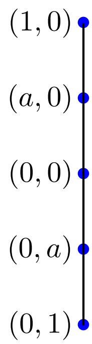
For $A \in \mathbf{PDL}$ and an existential quantifier $\exists$ on A, that is, $\exists$ satisfies (MB1)-(MB4), we can define on $K(A)$ the operator $\Omega_\exists\colon K(A) \to K(A)$ by

$$\Omega_\exists(a, b) = (\exists a, (\exists a)^*). \tag{6}$$

It was shown in [16] that $\Omega_\exists$ is a weak quantifier on the KAN-algebra $K(A)$.

On the other hand, given $\langle A, \exists, \forall \rangle \in \mathbf{MPDL}$, since $\exists$ is an existential quantifier on A, we can construct a weak quantifier $\Omega_\exists$ on $K(A)$ as defined in (6). However, this operator does not coincide with the operator $\exists_K$ defined on $K(A)$, as illustrated by the following example.

Example 5.1. *We consider the monadic pseudocomplemented distributive lattice* $\langle A, \exists, \forall \rangle$ *from Remark 3.2. Applying Kalman's construction yields the centered KAN-algebra* $K(A) = \{(0,1), (0,a), (0,0), (a,0), (1,0)\}$*, illustrated by the Hasse diagram below, together with the associated strong and weak negations shown in the table.*



(x,y)	$\sim (x,y)$	$\neg(x,y)$
$(1,0)$	$(0,1)$	$(0,1)$
$(a,0)$	$(0,a)$	$(0,a)$
$(0,0)$	$(0,0)$	$(1,0)$
$(0,a)$	$(a,0)$	$(1,0)$
$(0,1)$	$(1,0)$	$(1,0)$

Thus, taking into account the definitions given in 6 we obtain the operators $\Omega_\exists$ *and* $\exists_K$ *as shown in the following table:*

(x,y)	$\exists_K(x,y)$	$\Omega_\exists(x,y)$
$(1,0)$	$(1,0)$	$(1,0)$
$(a,0)$	$(1,0)$	$(1,0)$
$(0,0)$	$(0,0)$	$(0,1)$
$(0,a)$	$(0,a)$	$(0,1)$
$(0,1)$	$(0,1)$	$(0,1)$

5.2 Monteiro's construction

Monteiro's construction shows that, given a Nelson algebra, one can obtain a centered Nelson algebra. A similar process was established for KAN-algebras in [5]. In this subsection, we extend this result to the monadic setting.

Let $\langle T, \exists \rangle \in$ **MKAN**. Consider the set

$$M(T) := \{(x,y) \in T^{\Diamond} \times T^{\Box} : x \leq y\}$$

with the following operations on $M(T)$:

$$\begin{array}{rcl}(a,b)\cup(d,e) & := & (a\vee d,\Box(b\vee e)),\\ (a,b)\cap(d,e) & := & (\Diamond(a\wedge d),b\wedge e),\\ \boxplus(a,b) & := & (\sim b,\sim a),\\ \circledast(a,b) & := & (\Diamond\neg a,\sim a),\\ 0_M & := & (0,0),\\ 1_M & := & (1,1),\\ c_M & := & (0,1).\end{array}$$

By the results in [5], $\langle M(T),\cup,\cap,\boxplus,\circledast,c_M,0_M,1_M\rangle$ is a centered KAN-algebra. The order on $M(T)$ is given by $(a,b)\subseteq(d,e)$ if and only if $a\leq d$ and $b\leq e$. We define on $M(T)$ the following operators:

$$\exists_M(a,b)=(\exists a,\exists b)\ \text{ and }\ \forall_M(a,b)=(\forall a,\forall b).$$

It is easy to prove that $\forall_M(a,b)=\boxplus\exists_M\boxplus(a,b)$.

Theorem 5.3. *Let $\langle T,\exists\rangle\in$ **MKAN**. Then $\langle M(T),\exists_M\rangle\in$ **MKANc**. Moreover, $\langle T,\exists\rangle$ is isomorphic to a subalgebra of $\langle M(T),\exists_M\rangle$.*

Proof. Let $(a,b)\in M(T)$. Then $\Diamond a=a$, $\Box b=b$ and $a\leq b$. Hence, by Lemma 4.2, we have $\Diamond\exists a=\exists a$, $\Box\exists b=\exists b$ and $\exists a\leq\exists b$. Thus, $\exists_M(a,b)\in M(T)$. Therefore, $\exists_K$ is well-defined. Now, we proceed to prove that the conditions (MK1)–(MK8) are satisfied. To this end, let $(a,b),(d,e)\in M(T)$.

(MK1): By (MB1), $\exists_M(0,0)=(\exists 0,\exists 0)=(0,0)$.

(MK2): By (MB2), $(a,b)\subseteq(\exists a,\exists b)=\exists_M(a,b)$.

(MK3): From Lemma 4.2 (1) and (MB3) we have

$$\begin{array}{rcl}\exists_M((a,b)\cap\exists_M(d,e)) & = & \exists_M(\Diamond(a\wedge\exists d),b\wedge\exists e)\\ & = & (\exists\Diamond(a\wedge\exists d),\exists(b\wedge\exists e))\\ & = & (\Diamond(\exists a\wedge\exists d),\exists b\wedge\exists e)\\ & = & (\exists a,\exists b)\cap(\exists d,\exists e)\\ & = & \exists_M(a,b)\cap\exists_M(d,e).\end{array}$$

(MK4): By Lemma 4.2 (1) and (MB4),

$$\begin{array}{rcl}\exists_M((a,b)\cup(d,e)) & = & (\exists(a\vee d),\exists\Box(b\vee e))\\ & = & (\exists a\vee\exists d,\Box(\exists b\vee\exists e))\\ & = & (\exists a,\exists b)\cup(\exists d,\exists e)\\ & = & \exists_M(a,b)\cup\exists_M(d,e).\end{array}$$

(MK5): By (MK5) we have $\forall_M \exists_M(a,b) = (\forall\exists a, \forall\exists b) = (\exists a, \exists b) = \exists_M(a,b)$.

(MK6): From Lemma 4.2 (1) and (MB4) it follows that

$$\begin{array}{rcl}\forall_M(a,b) \cap \exists_M(d,e) & = & (\Diamond(\forall a \wedge \exists d), \forall b \wedge \exists e) \\ & \subseteq & (\Diamond\exists(a \wedge d), \exists(b \wedge e)) \\ & = & (\exists\Diamond(a \wedge d), \exists(b \wedge e)) \\ & = & \exists_M(\Diamond(a \wedge d), b \wedge e) \\ & = & \exists_M((a,b) \cap (d,e)).\end{array}$$

(MK7): By Lemma 4.2 (2) and (MK7),

$$\begin{array}{rcl}\forall_M \circledast (a,b) & = & (\forall\Diamond\neg a, \forall \sim a) \\ & = & (\Diamond\forall\neg a, \sim \exists a) \\ & = & (\Diamond\neg\exists a, \sim \exists a) \\ & = & \circledast\exists_M(a,b).\end{array}$$

(MK8): From Lemma 4.2 (1) and (MK8), we have

$$\begin{array}{rcl}\circledast\forall_M(a,b) & = & (\Diamond\neg\forall a, \sim \forall a) \\ & = & (\Diamond\exists\neg a, \exists \sim a) \\ & = & (\exists\Diamond\neg a, \exists \sim a) \\ & = & \exists_M \circledast (a,b).\end{array}$$

From the results in [5], we know that T is isomorphic to a subalgebra of $M(T)$ via the map $\delta\colon T \to M(T)$ given by $\delta(x) = (\Diamond x, \Box x)$. We now show that δ is compatible with the existential quantifier. By Lemma 4.2, we have

$$\delta(\exists x) = (\Diamond\exists x, \Box\exists x) = (\exists\Diamond x, \exists\Box x) = \exists_M(\Diamond x, \Box x) = \exists_M(\delta(x)).$$

Therefore, $\langle T, \exists\rangle$ is isomorphic to a subalgebra of $\langle M(T), \exists_M\rangle$. □

Let $\mathbb{MKAN}$ denote the category of monadic KAN-algebras where morphisms are the corresponding algebra homomorphisms. Notably, if $\langle T_1, \exists\rangle$ and $\langle T_2, \exists\rangle$ are centered monadic KAN-algebras and $f\colon T_1 \to T_2$ is a morphism in $\mathbb{MKAN}$, then f necessarily preserves the center. Indeed, we have $f(c) = f(\sim c) = \sim f(c)$ and thus, by the uniqueness of the center, $f(c) = c$. Therefore, $\mathbb{MKANC}$ is a full subcategory of $\mathbb{MKAN}$. Additionally, if $f\colon T_1 \to T_2$ is a morphism in $\mathbb{MKAN}$, then $M(f)\colon M(T_1) \to M(T_2)$ defined by $M(f)(x,y) = (f(x), f(y))$ for all $(x,y) \in M(T_1)$ is a morphism in $\mathbb{MKANC}$. By Theorem 5.3, M is a functor from the category $\mathbb{MKAN}$ to $\mathbb{MKANC}$.

Now, we explore the relationship between Kalman's construction and Monteiro's construction.

Theorem 5.4. *Let* $\langle T, \exists \rangle \in$ **MKAN**. *Then the centered monadic KAN-algebras* $\langle M(T), \exists_M \rangle$ *and* $\langle K(T^{\Diamond}), \exists_K^{\Diamond} \rangle$ *are isomorphic.*

Proof. By the results established in [5], $M(T)$ is isomorphic to $K(T^{\Diamond})$ via the map $t\colon M(T) \to K(T^{\Diamond})$ defined by $t(x,y) = (x, \sim y)$. It only remains to prove that t commutes with the existential quantifier. Let $(a,b) \in M(T)$. Then $\Diamond a = a$, $\Box b = b$, and $a \leq b$. Therefore, by Lemma 4.2,

$$\begin{array}{rcl} t(\exists_M(a,b)) & = & (\exists a, \sim \exists b) \\ & = & (\exists \Diamond a, \sim \exists \Box b) \\ & = & (\exists \Diamond a, \forall \sim \Box b) \\ & = & (\exists \Diamond a, \forall \Diamond \sim b) \\ & = & (\Diamond \exists a, \Diamond \forall \sim b) \\ & = & (\exists^{\Diamond} a, \forall^{\Diamond} \sim b) \\ & = & \exists_K^{\Diamond} t(a,b), \end{array}$$

i.e., $t(\exists_M(a,b)) = \exists_K^{\Diamond} t(a,b)$. □

Corollary 5.5. *Let* $\langle T, \exists \rangle \in$ **MKANc**. *Then the monadic centered KAN-algebras* $\langle T, \exists \rangle$ *and* $\langle M(T), \exists_M \rangle$ *are isomorphic.*

Proof. This follows from the results in [5] and Theorem 5.4. □

5.3 Sendlewski's construction

In order to generalize the construction given by Sendlewski in [24], the authors of [16] studied the variety of KAN-algebras and their connection with pairs consisting of a pseudocomplemented distributive lattice and a Boolean filter. They showed that if A is a pseudocomplemented distributive lattice and $F \subseteq A$ is a Boolean filter, then the set

$$K(A,F) := \{(a,b) \in A \times A \colon a \wedge b = 0 \text{ and } a \vee b \in F\}$$

endowed with the same operations as $K(A)$, constitutes the universe of an algebra in **KAN**. Conversely, given any $T \in$ **KAN**, there exists a congruence θ with respect to the operations $\vee$, $\wedge$, and $\neg$ such that T^{θ} is a pseudocomplemented distributive lattice and the quotient set T^{+}/θ, where

$$T^{+} := \{x \in T : \Box x = 1\}$$

is a Boolean filter (see [16]).

We denote by $\mathbb{PDLF}$ the category whose objects are pairs $\langle A, F\rangle$, where A is an object of the category **PDL** and F is a Boolean filter of A. The morphisms $f\colon \langle A, F\rangle \to \langle A', F'\rangle$ are morphisms $f\colon A \to A'$ in $\mathbb{PDL}$ such that $f(F) \subseteq F'$. Therefore, by the results given in [16], there is a categorical equivalence between the category $\mathbb{PDLF}$ and the category $\mathbb{KAN}$.

Denote by $\mathbb{MPDLF}$ the category whose objects are pairs consisting of an algebra $\langle A, \exists, \forall\rangle \in$ **MPDL** and a monadic Boolean filter F of A. The morphisms of $\mathbb{MPDLF}$ are the morphisms of $\mathbb{PDLF}$ that preserve the respective monadic operators.

Proposition 5.6. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL** and let $F \subseteq A$ be a monadic Boolean filter. Then $\langle K(A, F), \exists_K\rangle \in$ **MKAN**. Moreover, K extends to a functor from $\mathbb{MPDLF}$ to $\mathbb{MKAN}$.*

Proof. Let $(a, b) \in K(A, F)$. Then $a \wedge b = 0$ and $a \vee b \in F$. Since F is a monadic Boolean filter it follows $\forall(a \vee b) \in F$. By Lemma 4.1 (10) we have $\forall a \vee \exists b \in F$. Moreover, by axioms (MK6) and (MK1), we obtain $\forall a \wedge \exists b = 0$. Hence, $\exists_K(a, b) \in K(A, F)$ and therefore $\exists_K$ is well–defined. On the other hand, similarly to the proof of Theorem 5.1, it can be shown that $\exists_K$ satisfies the conditions (MK1)-(MK8). Therefore, it only remains to prove that K is functorial.

Let $f\colon \langle A, \exists, \forall, F\rangle \to \langle A', \exists', \forall', F'\rangle$ be a morphism in $\mathbb{MPDLF}$. For each $(a, b) \in K(A, F)$ we have

$$\begin{aligned} K(f)(\exists_K(a, b)) &= K(f)(\exists a, \forall b) \\ &= (f(\exists a), f(\forall b)) \\ &= (\exists' f(a), \forall' f(b)) \\ &= \exists'_K(f(a), f(b)) \\ &= \exists'_K(K(f)(a, b)), \end{aligned}$$

as desired. □

Remark 5.3. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL** and let $F \subseteq A$ be a monadic Boolean filter. Then, by Theorems 4.3 and 5.1, we have $\langle K(A, F)/\theta, \exists^\theta_K, \forall^\theta_K\rangle \in$ **MPDL**, where the monadic operators $\exists^\theta_K$ and $\forall^\theta_K$ are defined by*

$$\exists^\theta_K([(a, b)]_\theta) = [(\exists a, \forall b)]_\theta \text{ and } \forall^\theta_K([(a, b)]_\theta) = [(\forall a, \exists b)]_\theta.$$

Lemma 5.3. *Let $\langle A, \exists, \forall\rangle \in$ **MPDL** and let $F \subseteq A$ be a monadic Boolean filter. Then the map $g^{-1}\colon A \to K(A, F)/\theta$ defined by $g^{-1}(x) := [(x, x^*)]_\theta$ is an isomorphism.*

Proof. Taking into account the results of [16], it remains to show that g^{-1} preserves the monadic operators $\exists$ and $\forall$. Let $a \in A$. By Lemma 3.2 (1) we have

$$\begin{array}{rcl} g^{-1}(\exists a) & = & [(\exists a, (\exists a)^*)]_\theta \\ & = & [(\exists a, \forall a^*)]_\theta \\ & = & \exists_K^\theta [(a, a^*)]_\theta \\ & = & \exists_K^\theta g^{-1}(a), \end{array}$$

and by Lemma 3.2 (2),

$$\begin{array}{rcl} g^{-1}(\forall a) & = & [(\forall a, (\forall a)^*)]_\theta \\ & = & [(\forall a, \exists a^*)]_\theta \\ & = & \forall_K^\theta [(a, a^*)]_\theta \\ & = & \forall_K^\theta g^{-1}(a). \end{array}$$

Therefore, g^{-1} is an isomorphism. □

Lemma 5.4. *Let* $\langle T, \exists \rangle \in \mathbf{MKAN}$. *Then* $\langle K(T^\theta, T^+/\theta), \exists_K \rangle \in \mathbf{MKAN}$.

Proof. First we show that T^+/θ is a monadic Boolean filter. It is clear that T^+/θ is a Boolean filter (see [16, Lemma 2.14]). To prove that it is monadic, we must show that it is closed under the operator $\forall$. Let $x \in T^+$. Then $\Box x = 1$ and by Lemmas 4.2 and 4.1 it follows that $\Box \forall x = \forall \Box x = \forall 1 = 1$. Hence, $\forall x \in T^+$ and T^+/θ is a monadic Boolean filter. By Proposition 5.6 and Theorem 4.3 the result follows. □

Lemma 5.5. *Let* $\langle T, \exists \rangle \in \mathbf{MKAN}$. *Then the map* $\rho\colon T \to K(T^\theta, T^+/\theta)$ *defined by* $\rho(x) = ([x]_\theta,\ [\sim x]_\theta)$ *is an isomorphism.*

Proof. From the results of [16], it remains to show that ρ commutes with the operator $\exists$. Indeed,

$$\begin{array}{rcl} \rho(\exists x) & = & ([\exists x]_\theta,\ [\sim \exists x]_\theta) \\ & = & ([\exists x]_\theta,\ [\forall \sim x]_\theta) \\ & = & \left(\exists^\theta [x]_\theta,\ \forall^\theta [\sim x]_\theta\right) \\ & = & \exists_K^\theta ([x]_\theta,\ [\sim x]_\theta) \\ & = & \exists_K^\theta \rho(x), \end{array}$$

which completes the proof. □

Remark 5.4. *If* $T, T' \in \mathbf{MKAN}$ *and* $f\colon T \to T'$ *is a morphism in* $\mathbb{MKAN}$, *then* $f^\theta\colon T^\theta \to T'/\theta$ *given by* $f^\theta([x]_\theta) = [f(x)]_\theta$ *is a morphism in* $\mathbb{MPDLF}$.

Theorem 5.7. *The categories* $\mathbb{MPDLF}$ *and* $\mathbb{MKAN}$ *are categorically equivalent.*

Conclusions and future work

In this work, we have introduced and studied the variety of monadic pseudocomplemented distributive lattices, thereby extending the classical framework of monadic Boolean algebras to a broader non-Boolean context. We have established several fundamental properties, presented a Glivenko-type theorem, and described a categorical equivalence with classes of centered monadic KAN-algebras. Furthermore, we have generalized various classical constructions–such as those of Kalman, Monteiro, and Sendlewski–to the monadic setting, and we have clarified the relationships among these three constructions.

Future research may proceed along several directions. For example, it would be interesting to investigate the representation theory and dualities for monadic pseudocomplemented distributive lattices. Another promising line of research concerns the study of further classes of quantifiers or alternative monadic operators in related varieties, such as Stone algebras.

References

[1] Almiñana Reinoso F., Pelaitay G.: *Monadic $k \times j$-rough Heyting algebras.* Arch. Math. Logic **60** (2021), 923–949.

[2] Balbes R., Dwinger P.: *Distributive Lattices.* University of Missouri Press, Columbia, 1974.

[3] Bezhanishvili G.: *Varieties of monadic Heyting algebras. Part I.* Stud. Log. **61** (1998), 367–402.

[4] Calomino I.: *Monadic quasi-modal distributive nearlattices.* Commentat. Math. Univ. Carol. **64** (2023), 161–174.

[5] Calomino I., Pelaitay G.: *A new categorical equivalence for Stone algebras.* Accepted in Mathematica Slovaca (2025).

[6] Cignoli R.: *Quantifiers on distributive lattices.* Discrete Math. **96** (1991), 183–197.

[7] Chajda I., Halaš R., Kühr J.: *Semilattice Structures.* Heldermann Verlag, Research and Exposition in Mathematics, 2007.

[8] Davey B.: *Subdirectly irreducible distributive double p-algebras.* Algebra Univers. **8** (1978), 73–88.

[9] Di Nola A., Grigolia R.: *On monadic MV-algebras.* Ann. Pure Appl. Logic **128** (2004), 125–139.

[10] Figallo A.V., Gomes C.M.: *Monadic Boolean algebras with an automorphism and their relation to* $\mathbf{Df_2}$*-algebras.* Soft Comput. **24** (2020), 227–236.

[11] Figallo A.V., Pascual I.B.: *Topological dualities for strong monadic distributive lattices and applications.* J. Mult.-Val. Log. Soft Comput. **32** (2019), 499–540.

[12] Figallo A.V., Pascual I.B., Ziliani A.: *Monadic distributive lattices and monadic augmented Kripke frames*. J. Mult.-Val. Log. Soft Comput. **22** (2014), 189–216.

[13] Figallo A.V., Pascual I.B., Ziliani A.: *Monadic distributive lattices*. Log. J. IGPL **15** (2007), 535–551.

[14] Figallo Orellano, A. : *A preliminary study of MV-algebras with two quantifiers which commute*. Stud. Log. **104** (2016), 931–956.

[15] Ghorbani S.: *On monadic semi-Nelson algebras*. J. Appl. Log.-IfCoLog J. Log. Appl. **11** (2024), 625–643.

[16] Gomez C., Marcos M., San Martín H.J.: *On the relation of negations in Nelson algebras*. Rep. Math. Logic **56** (2021), 15–56.

[17] Halmos P.: *Algebraic logic I. Monadic Boolean algebras*. Compos. Math. **12** (1956), 217–249.

[18] Hecht T., Katriňák T.: *Principal congruences of p-algebras and double p-algebras*. Proc. Amer. Math. Soc. **58** (1976), 25–31.

[19] Katriňák T.: *Free distributive p-algebras: a new approach*. Glasgow Math. J. **40** (1998), 333–342.

[20] Kalman J.A.: *Lattices with involution*. Trans. Amer. Math. Soc. **87** (1958), 485–491.

[21] Lakser H.: *The structure of pseudocomplemented distributive lattices. I. Subdirect decomposition*. Trans. Amer. Math. Soc. **156** (1971), 334–342.

[22] Rao M.S., Shum K.: *Boolean filters of distributive lattices*. International Journal of Math. and Soft Comput. **3** (2013), 41–48.

[23] Schmid J.: *Algebraically closed distributive p-algebras*. Algebra Univers. **15** (1982), 126–141.

[24] Sendlewski A.: *Nelson algebras through Heyting ones. I*. Stud. Log. **49** (1990), 105–126.

[25] Sendlewski A.: *Topologicality of Kleene algebras with a weak pseudocomplementation over distributive p-algebras*. Rep. Math. Logic **25** (1991), 13–56.

[26] Vakarelov D.: *Notes on N-lattices and constructive logic with strong negation*. Stud. Log. **36** (1977), 109–125.

[27] Wang J., He P., She Y.: *Monadic NM-algebras*. Log. J. IGPL **27** (2019), 812–835.

[28] Wang J., He P., Yang J., Wang M., He X.: *Monadic NM-algebras: an algebraic approach to monadic predicate nilpotent minimum logic*. J. Logic Comput. **32** (2022), 741–766.

[29] Wang J.T., She Y.H., He P.F., Ma N.N.: *On categorical equivalence of weak monadic residuated distributive lattices and weak monadic c-differential residuated distributive lattices*. Stud. Log. **111** (2023), 361–390.

[30] Wang J., Wu H., He P., She Y.: *An algebraic proof of completeness for monadic fuzzy predicate logic MMTL∀*. Rev. Symb. Log. **18** (2025), 213–239.

Received 13 June 2025