# AXIOMATIZATION OF IF-THEN-ELSE OVER POSSIBLY NON-HALTING PROGRAMS AND TESTS

GAYATRI PANICKER, K. V. KRISHNA, AND PURANDAR BHADURI

Abstract. In order to study the axiomatization of the `if-then-else` construct over possibly non-halting programs and tests, this paper introduces the notion of *C-sets* by considering the tests from an abstract $C$-algebra. When the $C$-algebra is an ada, the axiomatization is shown to be complete by obtaining a subdirect representation of $C$-sets. Further, this paper considers the equality test with the `if-then-else` construct and gives a complete axiomatization through the notion of *agreeable C-sets*.

Communicated by Mikhail Volkov

## Introduction

Being one of the fundamental constructs, the conditional expression `if-then-else` has received considerable importance in programming languages. It plays a vital role in the study of program semantics. One of the seminal works in the axiomatization of this conditional expression was by McCarthy in [19], where he gave an axiom schema for the determination of the semantic equivalence between any two conditional expressions. Since then several authors have studied the axiomatization of `if-then-else` in different contexts.

Following McCarthy's approach, Igarashi in [10] studied a formal system comprising ALGOL-like statements including various programming features along with `if-then-else` with predicates. The two systems were shown to be equivalent by de Bakker in [6], i.e., axioms of one could be derived from the other. In [22] Sethi gave a different framework to determine the semantic equivalence of statements of the form `if` $E = F$ `then` $G$ `else` $H$. In [13] Kennison defined comparison algebras as those equipped with a quaternary operation $C(s,t,u,v)$ satisfying certain identities modelling the equality test. He also showed that such algebras are simple if and only if $C$ is the direct comparison operation $C_0$ given by $C_0(s,t,u,v)$ taking value $u$ if $s = t$ and $v$ otherwise. In [21] Pigozzi gave an axiomatization of the theory of equality test algebras appended with `if-then-else`, where the test is purely $T$ (`true`) or $F$ (`false`). He gave a finite axiom scheme for the quasi-equational theory of equality test algebras and another finite axiom scheme for the equational theory of `if-then-else` algebras. In [2] Bergman studied the sheaf-theoretic representation of sets equipped with an action of a Boolean algebra. This Boolean action was in fact the `if-then-else` function. This approach was adopted in [24] by Stokes who obtained a representation theorem for the Boolean algebra case of

2010 *Mathematics Subject Classification.* 08A70, 03G25 and 68N15.
*Key words and phrases.* Axiomatization, if-then-else, non-halting programs, $C$-algebra.

`if-then-else` algebras of [17]. Further in [23] Stokes extended the work of Kennison to semigroups and monoids. He showed that every comparison semigroup (monoid) is embeddable in the comparison semigroup (monoid) $\mathcal{T}(X)$ of all total functions $X \to X$, for some set $X$. He also obtained a similar result in terms of partial functions $X \to X$.

In [11] Jackson and Stokes gave a complete axiomatization of `if-then-else` over halting programs and tests. They also modelled composition of functions and of functions with predicates and further showed that the more natural setting of only considering composition of functions would not admit a finite axiomatization.

The work listed above mainly focus on halting tests (by assuming them to be of Boolean type) and halting programs. A natural interest in this context is to study non-halting tests and programs. Along these lines as well considerable work has been done, besides the work in [19] and [10].

In [4] Bloom and Tindell studied four versions of `if-then-else` along with the equality test. In two cases they considered the halting scenario whilst in the other two they modelled possibly non-halting programs and tests. They provided an equationally complete proof system for each such framework while noting that none of the classes formed an equational class. In order to obtain similar results in the context of functional programming languages that have user-definable data types, in [7], Guessarian and Meseguer extended the proof system of [4] to heterogeneous algebras that have extra operations, predicates and equations. Another extension of [4] was by Mekler and Nelson in [20]. In this work the authors expanded the algebras in some equational class $K$ by adding the `if-then-else` operation and found axioms for the equational class $K^*$ generated by these algebras. They also showed that the equational theory for $K^*$ is decidable if the word problem for $K$ is decidable. On a slightly different track, Manes in [18] gave a transformational characterisation of `if-then-else` where the tests are Boolean but the functions on which they act could be non-halting. Further, in [17], Manes considered `if-then-else` algebras over Boolean algebras, $C$-algebras and adas. Here $C$-algebras and adas are algebras of non-halting tests, generalizing Boolean algebras to three-valued logics.

While there are several studies (cf. [1], [3], [5], [9], [14], [16]) on extending two-valued Boolean logic to three-valued logic, McCarthy's logic (cf. [19]) models the lazy evaluation exhibited by programming languages that evaluate expressions in sequential order, from left to right. In [8] Guzmán and Squier gave a complete axiomatization of McCarthy's three-valued logic and called the corresponding algebra a $C$-algebra, or the algebra of conditional logic. While studying `if-then-else` algebras in [17], Manes defined an *ada* (Algebra of Disjoint Alternatives) which is essentially a $C$-algebra equipped with an oracle for the halting problem.

Recently, in [12] Jackson and Stokes studied the algebraic theory of computable functions, which can be viewed as possibly non-halting programs, equipped with composition, `if-then-else` and `while-do`. In this work they assumed that the tests form a Boolean algebra. Further, they demonstrated how an algebra of non-halting tests could be constructed from Boolean tests in their setting. Jackson and Stokes proposed an alternative approach by considering an abstract collection of non-halting tests as in [17] and posed the following problem:

*Characterize the algebras of computable functions associated with an abstract $C$-algebra of non-halting tests.*

In this paper, we attempt to address the problem by adopting the approach of Jackson and Stokes in [11]. To this end, we define the notion of a *C-set* through which we provide a complete axiomatization for `if-then-else` over a class of possibly non-halting programs and tests, where tests are drawn from an ada. The paper has been organised as follows. The necessary background material is provided in Section 1. In Section 2, we introduce the notion of $C$-sets and give a few properties of $C$-sets. Section 3 is dedicated to providing a subdirect representation of $C$-sets over adas. Further, in Section 4 we give a complete axiomatization for $C$-sets over adas equipped with the equality test, called *agreeable C-sets*. A brief conclusion with possible extensions of this work is presented in Section 5.

## 1. Preliminaries

In this section, we list definitions and results that will be useful to us. In [11] Jackson and Stokes considered the notion of a $B$-set, which was introduced by Bergman in [2], in order to study the theory of halting programs equipped with the operation of `if-then-else`.

**Definition 1.1.** Let $\langle Q, \vee, \wedge, \neg, T, F\rangle$ be a Boolean algebra and $S$ be a set. A *$B$-set* is a pair $(S, Q)$, equipped with a function, called *$B$-action* $\eta : Q \times S \times S \to S$, where $\eta(\alpha, a, b)$ is denoted by $\alpha[a, b]$, read "`if` $\alpha$ `then` $a$ `else` $b$", that satisfies the following axioms for all $\alpha, \beta \in Q$ and $a, b, c \in S$:

$$\alpha[a, a] = a \tag{1}$$

$$\alpha[\alpha[a, b], c] = \alpha[a, c] \tag{2}$$

$$\alpha[a, \alpha[b, c]] = \alpha[a, c] \tag{3}$$

$$F[a, b] = b \tag{4}$$

$$\neg\alpha[a, b] = \alpha[b, a] \tag{5}$$

$$(\alpha \wedge \beta)[a, b] = \alpha[\beta[a, b], b] \tag{6}$$

We recall the following examples from [11].

**Example 1.2.** For any Boolean algebra $Q$, the pair $(Q, Q)$ is a $B$-set with the following action for all $\alpha, \beta, \gamma \in Q$:

$$\alpha[\beta, \gamma] = (\alpha \wedge \beta) \vee (\neg\alpha \wedge \gamma).$$

**Example 1.3.** Consider the two-element Boolean algebra $\mathbb{2}$ with the universe $\{T, F\}$. For any set $S$, the pair $(S, \mathbb{2})$ is a $B$-set with the following action for all $a, b \in S$:

$$T[a, b] = a,$$
$$F[a, b] = b.$$

These $B$-sets are called *basic* $B$-sets.

**Notation 1.4.** Let $X$ and $Y$ be two sets. The set of all functions from $X$ to $Y$ will be denoted by $Y^X$. The set of all total functions $X \to X$ will be denoted by $\mathcal{T}(X)$.

**Example 1.5.** For any set $X$, the pair $(\mathcal{T}(X), 2^X)$ is a $B$-set with the following action for all $\alpha \in 2^X$ and $g, h \in \mathcal{T}(X)$:

$$\alpha[g,h](x) = \begin{cases} g(x), & \text{if } \alpha(x) = T; \\ h(x), & \text{if } \alpha(x) = F. \end{cases}$$

In [11] Jackson and Stokes showed that every $B$-set can be represented in terms of basic $B$-sets.

**Theorem 1.6** ([11]). *Every $B$-set is a subdirect product of basic $B$-sets.*

This tells us that studying the identities or quasi-identities satisfied by the subclass of basic $B$-sets suffices to understand those satisfied by the entire class of $B$-sets. Checking the validity of any identity (quasi-identity) in a basic $B$-set involves merely checking the respective values for `true` and `false` and is thus far simpler than checking the same in an arbitrary $B$-set. Further, in [11], they model the equality test based on the assumption that the tests arise from a Boolean algebra and that the functions are halting.

**Definition 1.7.** A $B$-set $(S, B)$ is said to be *agreeable* if it is equipped with an operation $* : S \times S \to B$ satisfying the following axioms for all $s, t, u, v \in S$ and $\alpha \in B$:

$$s * s = T \tag{7}$$

$$(s * t)[s, t] = t \tag{8}$$

$$\alpha[s, t] * \alpha[u, v] = \alpha[s * u, t * v] \tag{9}$$

The following are examples of agreeable $B$-sets.

**Example 1.8.** The pair $(\mathcal{T}(X), 2^X)$ is an agreeable $B$-set with the operation $*$ defined as follows for all $f, g \in \mathcal{T}(X)$:

$$(f * g)(x) = \begin{cases} T, & \text{if } f(x) = g(x); \\ F, & \text{otherwise.} \end{cases}$$

**Example 1.9.** Let $S$ be any set. The pair $(S, 2)$ is an agreeable $B$-set under the operation $*$ defined in the following manner for all $s, t \in S$:

$$s * t = \begin{cases} T, & \text{if } s = t; \\ F, & \text{otherwise.} \end{cases}$$

These $B$-sets are called *basic agreeable $B$-sets.*

Jackson and Stokes proved the following result.

**Theorem 1.10** ([11]). *Every agreeable $B$-set is a subdirect product of basic agreeable $B$-sets.*

In [15] Kleene discussed various three-valued logics that are extensions of Boolean logic. McCarthy first studied the three-valued non-commutative logic in the context of programming languages in [19]. This is the non-commutative regular extension of Boolean logic to three truth values. Here the third truth value $U$ denotes the `undefined` state which is attained when a test diverges. In this new context, the evaluation of expressions is carried out sequentially from left to right, mimicking that of a majority of programming languages. The complete axiomatization for the class of algebras associated with this logic was given by Guzmán and Squier in [8]. They called the algebra associated with this logic a *C-algebra*. We shall denote an arbitrary $C$-algebra by $M$.

**Definition 1.11.** A *C-algebra* is an algebra $\langle M, \vee, \wedge, \neg\rangle$ of type $(2,2,1)$, which satisfies the following axioms for all $\alpha, \beta, \gamma \in M$:

$$\neg\neg\alpha = \alpha \tag{10}$$

$$\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta \tag{11}$$

$$(\alpha \wedge \beta) \wedge \gamma = \alpha \wedge (\beta \wedge \gamma) \tag{12}$$

$$\alpha \wedge (\beta \vee \gamma) = (\alpha \wedge \beta) \vee (\alpha \wedge \gamma) \tag{13}$$

$$(\alpha \vee \beta) \wedge \gamma = (\alpha \wedge \gamma) \vee (\neg\alpha \wedge \beta \wedge \gamma) \tag{14}$$

$$\alpha \vee (\alpha \wedge \beta) = \alpha \tag{15}$$

$$(\alpha \wedge \beta) \vee (\beta \wedge \alpha) = (\beta \wedge \alpha) \vee (\alpha \wedge \beta) \tag{16}$$

**Example 1.12.** Every Boolean algebra is a $C$-algebra. In particular, $\mathbb{2}$ is a $C$-algebra.

**Example 1.13.** Let $\mathbb{3}$ denote the $C$-algebra with the universe $\{T, F, U\}$ and the following operations. This is, in fact, McCarthy's three-valued logic.

| $\neg$ | |
|---|---|
| $T$ | $F$ |
| $F$ | $T$ |
| $U$ | $U$ |

| $\wedge$ | $T$ | $F$ | $U$ |
|---|---|---|---|
| $T$ | $T$ | $F$ | $U$ |
| $F$ | $F$ | $F$ | $F$ |
| $U$ | $U$ | $U$ | $U$ |

| $\vee$ | $T$ | $F$ | $U$ |
|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $F$ | $U$ |
| $U$ | $U$ | $U$ | $U$ |

In view of the fact that the class of $C$-algebras is a variety, for any set $X$, $\mathbb{3}^X$ is a $C$-algebra with the operations defined pointwise. In fact, in [8], Guzmán and Squier showed that elements of $\mathbb{3}^X$ along with the $C$-algebra operations may be viewed in terms of *pairs of sets*. This is a pair $(A, B)$ where $A, B \subseteq X$ and $A \cap B = \emptyset$. Akin to the well-known correlation between $\mathbb{2}^X$ and the power set $\wp(X)$ of $X$, for any element $\alpha \in \mathbb{3}^X$, associate the pair of sets $(A, B)$ where $A = \{x \in X : \alpha(x) = T\}$ and $B = \{x \in X : \alpha(x) = F\}$. Conversely, for any pair of sets $(A, B)$ where $A, B \subseteq X$ and $A \cap B = \emptyset$ associate the function $\alpha$ where $\alpha(x) = T$ if $x \in A$, $\alpha(x) = F$ if $x \in B$ and $\alpha(x) = U$ otherwise. With this correlation, the operations can be expressed as follows:

$$\neg(A_1, A_2) = (A_2, A_1)$$
$$(A_1, A_2) \wedge (B_1, B_2) = (A_1 \cap B_1, A_2 \cup (A_1 \cap B_2))$$
$$(A_1, A_2) \vee (B_1, B_2) = ((A_1 \cup (A_2 \cap B_1), A_2 \cap B_2)$$

Further, Guzmán and Squier showed that every $C$-algebra is a subalgebra of $\mathbb{3}^X$ for some $X$ as stated below.

**Theorem 1.14** ([8]). *$\mathbb{3}$ and $\mathbb{2}$ are the only subdirectly irreducible $C$-algebras. Hence, every $C$-algebra is a subalgebra of a product of copies of $\mathbb{3}$.*

*Remark* 1.15. Considering a $C$-algebra $M$ as a subalgebra of $\mathbb{3}^X$, one may observe that $M_\# = \{\alpha \in M : \alpha \vee \neg\alpha = T\}$ forms a Boolean algebra under the induced operations.

**Notation 1.16.** A $C$-algebra with $T, F, U$ is a $C$-algebra with nullary operations $T, F, U$, where $T$ is the (unique) left-identity (and right-identity) for $\wedge$, $F$ is the (unique) left-identity (and right-identity) for $\vee$ and $U$ is the (unique) fixed point for $\neg$. Note that $U$ is also a left-zero for both $\wedge$ and $\vee$ while $F$ is a left-zero for $\wedge$.

There is an important subclass of the variety of $C$-algebras. In [17] Manes introduced the notion of *ada* (algebra of disjoint alternatives) which is a $C$-algebra equipped with an oracle for the halting problem. He showed that the category of adas is equivalent to that of Boolean algebras. The $C$-algebra $\mathbb{3}$ is not functionally-complete. However, $\mathbb{3}$ is functionally-complete when treated as an ada. In fact, the variety of adas is generated by the ada $\mathbb{3}$.

**Definition 1.17.** An *ada* is a $C$-algebra $M$ with $T, F, U$ equipped with an additional unary operation $(\ )^\downarrow$ subject to the following equations for all $\alpha, \beta \in M$:

$$F^\downarrow = F \tag{17}$$

$$U^\downarrow = F \tag{18}$$

$$T^\downarrow = T \tag{19}$$

$$\alpha \wedge \beta^\downarrow = \alpha \wedge (\alpha \wedge \beta)^\downarrow \tag{20}$$

$$\alpha^\downarrow \vee \neg(\alpha^\downarrow) = T \tag{21}$$

$$\alpha = \alpha^\downarrow \vee \alpha \tag{22}$$

**Example 1.18.** The three-element $C$-algebra $\mathbb{3}$ with the unary operation $(\ )^\downarrow$ defined as follows forms an ada.

$$T^\downarrow = T$$
$$U^\downarrow = F = F^\downarrow$$

We shall also use $\mathbb{3}$ to denote this ada. One may easily resolve the notation overloading – whether $\mathbb{3}$ is a $C$-algebra or an ada – depending on the context.

In [17] Manes showed that the three-element ada $\mathbb{3}$ is the only subdirectly irreducible ada. For any set $X$, $\mathbb{3}^X$ is an ada with operations defined pointwise. Note that the three element ada $\mathbb{3}$ is also simple. Manes also showed the following result.

**Proposition 1.19** ([17])**.** *Let $A$ be an ada. Then $A^\downarrow = \{\alpha^\downarrow : \alpha \in A\}$ forms a Boolean algebra under the induced operations.*

*Remark* 1.20. In fact, $A^\downarrow = A_\#$. Also, $A^\downarrow = \{\alpha \in A : \alpha^\downarrow = \alpha\}$.

Further, as outlined in the following remark, Manes established that the category of adas and the category of Boolean algebras are equivalent.

*Remark* 1.21 ([17]). Let $B$ be a Boolean algebra. By Stone's representation of Boolean algebras, suppose $B$ is a subalgebra of $\mathbb{2}^X$ for some set $X$. Consider the subalgebra $B^\star$ of the ada $\mathbb{3}^X$ with the universe $B^\star = \{(P, Q) : P \cap Q = \emptyset\}$ given in terms of pairs of subsets of $X$. Note that the map $B \mapsto (B^\star)_\#$ is a Boolean isomorphism. Similarly, for an ada $A$, the map $A \mapsto (A_\#)^\star$ is an ada isomorphism. Hence, the functor based on the aforesaid assignment establishes that the category of adas and the category of Boolean algebras are equivalent.

**Notation 1.22.** Let $X$ be a set and $\perp \notin X$. The pointed set $X \cup \{\perp\}$ with base point $\perp$ is denoted by $X_\perp$. The set of all total functions on $X_\perp$ which fix $\perp$ is denoted by $\mathcal{T}_o(X_\perp)$, i.e. $\mathcal{T}_o(X_\perp) = \{f \in \mathcal{T}(X_\perp) \,:\, f(\perp) = \perp\}$.

## 2. $C$-sets

In this section we introduce the notion of a $C$-set to study an axiomatization of `if-then-else` that includes the models of possibly non-halting programs and tests. The concept of $C$-sets is an extension of that of $B$-sets, wherein the tests are drawn from a $C$-algebra instead of a Boolean algebra, and includes a non-halting or `error` state.

**Definition 2.1.** Let $S_\perp$ be a pointed set with base point $\perp$ and $M$ be a $C$-algebra with $T, F, U$. The pair $(S_\perp, M)$ equipped with an action

$$\_\,[\_\,,\_] : M \times S_\perp \times S_\perp \to S_\perp$$

is called a *$C$-set* if it satisfies the following axioms for all $\alpha, \beta \in M$ and $s, t, u, v \in S_\perp$:

$$U[s,t] = \perp \qquad (U\text{-axiom}) \tag{23}$$

$$F[s,t] = t \qquad (F\text{-axiom}) \tag{24}$$

$$(\neg\alpha)[s,t] = \alpha[t,s] \qquad (\neg\text{-axiom}) \tag{25}$$

$$\alpha[\alpha[s,t],u] = \alpha[s,u] \qquad (\text{positive redundancy}) \tag{26}$$

$$\alpha[s,\alpha[t,u]] = \alpha[s,u] \qquad (\text{negative redundancy}) \tag{27}$$

$$(\alpha \wedge \beta)[s,t] = \alpha[\beta[s,t],t] \qquad (\wedge\text{-axiom}) \tag{28}$$

$$\alpha[\beta[s,t],\beta[u,v]] = \beta[\alpha[s,u],\alpha[t,v]] \qquad (\text{premise interchange}) \tag{29}$$

$$\alpha[s,t] = \alpha[t,t] \Rightarrow (\alpha \wedge \beta)[s,t] = (\alpha \wedge \beta)[t,t] \qquad (\wedge\text{-compatibility}) \tag{30}$$

*Remark* 2.2. In view of equations (10) and (11) of $C$-algebras and (25) and (28) of $C$-sets, we have the following property in $C$-sets.

$$(\alpha \vee \beta)[s,t] = \alpha[s,\beta[s,t]] \qquad (\vee\text{-axiom}) \tag{31}$$

We now present the intuition behind the notion of $C$-set and its axioms with respect to program constructs. In order to include the possibility of non-halting tests, we assume that the tests form a $C$-algebra. A test diverges at a given input if the output evaluates to $U$, `undefined`. When a test diverges or if the program throws up an `error` or does not halt, we shall say that the program evaluates to $\perp$. Thus a pointed set $S_\perp$ models the set of states and base point $\perp$ serves to denote the `error` state.

The $U$-axiom (23) essentially encapsulates the real-world requirement that if a test diverges, the output should be the `error` state. The $F$-axiom (24) is natural since when the test is `false`, the `then` part of the `if-then-else` construct is executed. The $\neg$-axiom (25) simply states that executing `if` *not* $P$ `then` $f$ `else` $g$ is the same as executing `if` $P$ `then` $g$ `else` $f$. The axioms of positive redundancy and negative redundancy (26) and (27) encapsulate the *cascading* nature of `if-then-else`. The $\wedge$-axiom (28) states that evaluating the test $P$ *AND* $Q$ and then executing $f$ else $g$ works in exactly the same way as evaluating $P$ first, which if `true`, executing `if` $Q$ `then` $f$ `else` $g$, and if `false` then simply executing $g$. The axiom of premise interchange (29) serves as a *switching* law. This states that the behaviour of the program where $P$ is evaluated first and $Q$ is a test situated within both the branches of the main `if-then-else`, is exactly the same as evaluating $Q$ first with $P$ in each branch, on suitably interchanging the programs situated at the leaves. The last axiom of $\wedge$-compatibility (30) loosely means that if $f$ and $g$ agree with regards to some domain, then they will agree on any subdomain.

**Example 2.3.** Let $M$ be a $C$-algebra with $T, F, U$. By treating $M$ as a pointed set with base point $U$, the pair $(M, M)$ is a $C$-set under the following action for all $\alpha, \beta, \gamma \in M$:

$$\alpha[\beta, \gamma] = (\alpha \wedge \beta) \vee (\neg\alpha \wedge \gamma).$$

Hereafter, the action of the $C$-set $(M, M)$ will be denoted by double brackets $\_ [\![ \_ , \_ ]\!]$. For verification of the axioms (23) – (30) refer to Appendix A.1.

We now present the motivating example of $C$-sets. Since the natural models of possibly non-halting programs are partial functions, we consider the model $\mathcal{T}_o(X_\perp)$ in view of the following one-to-one correspondence between $\mathcal{T}_o(X_\perp)$ and the set of partial functions on a set $X$. Each partial function $f$ on $X$ is represented by the total function $f' \in \mathcal{T}_o(X_\perp)$ where $f'(x) = f(x)$ when $x$ is in the domain of $f$, and maps to $\perp$ otherwise. Conversely, each $g \in \mathcal{T}_o(X_\perp)$ is represented by the partial function $g''$ over $X$ where $g''(x) = g(x)$ when $x \in X$ and $g(x) \neq \perp$, and is not defined elsewhere. The model $\mathcal{T}_o(X_\perp)$ can be seen to be a $C$-set under the action of the $C$-algebra $\mathbb{3}^X$ as shown in the following example.

**Example 2.4.** Consider $\mathcal{T}_o(X_\perp)$ as a pointed set with base point $\zeta_\perp$, the constant function taking the value $\perp$. The pair $\big(\mathcal{T}_o(X_\perp), \mathbb{3}^X\big)$ is a $C$-set with the following action for all $f, g \in \mathcal{T}_o(X_\perp)$ and $\alpha \in \mathbb{3}^X$:

$$\alpha[f,g](x) = \begin{cases} f(x), & \text{if } \alpha(x) = T; \\ g(x), & \text{if } \alpha(x) = F; \\ \perp, & \text{otherwise.} \end{cases} \tag{32}$$

Note that the execution of the first two cases, $\alpha(x) \in \{T, F\}$ demands that $x \in X$ as $\alpha \in \mathbb{3}^X$. These $C$-sets will be called *functional $C$-sets*. For verification of the axioms (23) – (30) refer to Appendix A.2.

**Example 2.5.** Consider $S_\perp^X$, the set of all functions from $X$ to $S_\perp$, as a pointed set with base point $\zeta_\perp$. The pair $\big(S_\perp^X, \mathbb{3}^X\big)$ is a $C$-set under the action given in (32), where $f, g \in S_\perp^X$ and $\alpha \in \mathbb{3}^X$. The axioms (23) – (30) can be verified along the same lines as in Example 2.4.

**Example 2.6.** Consider $\mathcal{T}(X_\perp)$, the set of all total functions on $X_\perp$, as a pointed set with base point $\zeta_\perp$. The pair $\big(\mathcal{T}(X_\perp), \mathbb{3}^X\big)$ is a $C$-set under the action given in (32), where $f, g \in \mathcal{T}(X_\perp)$ and $\alpha \in \mathbb{3}^X$. The axioms (23) – (30) can be verified along the same lines as in Example 2.4.

We believe that the $C$-set given in Example 2.6 does not occur naturally in the context of programs as this would include elements that terminate even when the input diverges, i.e. the input is $\perp$.

We now present a fundamental example of a $C$-set, where we only consider the basic tests, `true, false, undefined`.

**Example 2.7.** Let $S_\perp$ be a pointed set with base point $\perp$. The pair $(S_\perp, \mathbb{3})$ is a $C$-set with respect to the following action for all $a, b \in S_\perp$ and $\alpha \in \mathbb{3}$:

$$\alpha[a,b] = \begin{cases} a, & \text{if } \alpha = T; \\ b, & \text{if } \alpha = F; \\ \perp, & \text{if } \alpha = U. \end{cases}$$

These $C$-sets are called *basic $C$-sets.* While the axioms (23) and (24) are easy to observe, the axioms (25) – (30) can be verified by considering $\alpha$ to be $T$, $F$ and $U$ casewise.

Henceforth, unless explicitly mentioned otherwise, an arbitrary $C$-set is always denoted by $(S_\perp, M)$. In the remainder of this section, we shall prove certain properties of $C$-sets.

**Proposition 2.8.** *The following statements hold for all $\alpha, \beta \in M$ and $s, t, r \in S_\perp$:*

(i) $\alpha[\perp, \perp] = \perp$.
(ii) *If $\alpha[s,u] = \alpha[t,q]$ for some $u, q \in S_\perp$ then $\alpha[s,v] = \alpha[t,v]$ for all $v \in S_\perp$.*
(iii) *If $\alpha[s,u] = \alpha[r,r]$ for some $u \in S_\perp$ then $\alpha[s,r] = \alpha[r,r]$.*
(iv) *If $\alpha[s,u] = \alpha[t,u]$ for some $u \in S_\perp$ then $\alpha[s,v] = \alpha[t,v]$ for all $v \in S_\perp$.*
(v) *If $\alpha[s,t] = \alpha[t,t]$ then $(\beta \wedge \alpha)[s,t] = (\beta \wedge \alpha)[t,t]$.*

*Proof.*

(i) Using (23) and (29), $\alpha[\perp,\perp] = \alpha[U[\perp,\perp], U[\perp,\perp]] = U[\alpha[\perp,\perp], \alpha[\perp,\perp]] = \perp$.
(ii) Using (26), $\alpha[s,v] = \alpha[\alpha[s,u],v] = \alpha[\alpha[t,q],v] = \alpha[t,v]$.
(iii) Using Proposition 2.8(ii), putting $t = q = v = r$, $\alpha[s,r] = \alpha[r,r]$.
(iv) Using Proposition 2.8(ii), putting $q = u$, $\alpha[s,v] = \alpha[t,v]$.
(v) Using (28), $(\beta \wedge \alpha)[s,t] = \beta[\alpha[s,t],t] = \beta[\alpha[t,t],t] = (\beta \wedge \alpha)[t,t]$.

□

*Remark* 2.9.

(i) The $C$-set axioms from (24) to (28) are the same as the ones in the definition of $B$-set. In view of (23), the only $B$-set axiom that does not carry over in the context of $C$-sets is (1).
(ii) Bergman in [2] showed that the axiom of premise interchange (29) holds in $B$-sets.
(iii) Following the proof given in Proposition 2.8(v) and using the commutativity of $\wedge$ in the context of $B$-sets, it can be observed that the axiom of $\wedge$-compatibility (30) holds in $B$-sets.

**Proposition 2.10.** *For each $\alpha \in M_\#$ and $s \in S_\perp$, we have $\alpha[s,s] = s$.*

*Proof.* Let $\alpha \in M_\#$ and $s \in S_\perp$.

$$\begin{aligned} s &= T[s,s] && \text{from } (25), (24) \\ &= (\alpha \vee (\neg\alpha))[s,s] && \text{since } \alpha \in M_\# \\ &= \alpha[s, (\neg\alpha)[s,s]] && \text{from } (31) \\ &= \alpha[s, \alpha[s,s]] && \text{from } (25) \\ &= \alpha[s,s] && \text{from } (27) \end{aligned}$$

□

In view of Proposition 2.10, the axiom (1) of $B$-sets holds for the elements of Boolean algebra $M_{\#}$. Hence, we have the following corollary.

**Corollary 2.11.** *The pair* $(S_\perp, M_{\#})$ *is a* $B$*-set.*

*Remark* 2.12. The proof of Proposition 2.10 also shows us that axiom (1) is redundant in the definition of a $B$-set.

## 3. Representation of $C$-sets

With the aim of studying structural properties of $C$-sets, in this section we obtain a subdirect representation of $C$-sets in which the $C$-algebras are adas. Note that, except in Example 2.3, the $C$-algebras in all other examples of $C$-sets given in Section 2 are adas.

Let $(S_\perp, M)$ be a $C$-set, where $M$ is an ada. In the main theorem (Theorem 3.13) of this section, we obtain a subdirect representation of $(S_\perp, M)$ through various results presented hereafter. In these results, we consistently use $\alpha, \beta, \gamma$ for the elements of $M$, and $r, s, t, u, v$ for the elements of $S_\perp$.

**Proposition 3.1.** *If* $\alpha[s,t] = \alpha[t,t]$*, then* $\alpha^\downarrow[s,t] = \alpha^\downarrow[t,t]$*.*

*Proof.* Using (22) and (31), $\alpha[s,t] = (\alpha^\downarrow \vee \alpha)[s,t] = \alpha^\downarrow[s, \alpha[s,t]] = \alpha^\downarrow[s, \alpha[t,t]]$. On the other hand, observe that $\alpha[s,t] = \alpha[t,t] = (\alpha^\downarrow \vee \alpha)[t,t] = \alpha^\downarrow[t, \alpha[t,t]]$ so that $\alpha^\downarrow[s, \alpha[t,t]] = \alpha^\downarrow[t, \alpha[t,t]]$. Consequently, we have $\alpha^\downarrow[s,t] = \alpha^\downarrow[t,t]$ by Proposition 2.8(iv). □

Considering the $C$-sets as two-sorted algebras, we now define congruences on them.

**Definition 3.2.** A *congruence* of a $C$-set is a pair $(\sigma, \tau)$, where $\sigma$ is an equivalence relation on $S_\perp$ and $\tau$ is a congruence on the ada $M$ such that

$$(s,t), (u,v) \in \sigma \ \text{ and } \ (\alpha, \beta) \in \tau \Rightarrow (\alpha[s,u], \beta[t,v]) \in \sigma.$$

**Notation 3.3.** Under an equivalence relation $\sigma$ on a set $A$, the equivalence class of an element $p \in A$ will be denoted by $\overline{p}^\sigma$. Within a given context, if there is no ambiguity, we may simply denote the equivalence class by $\overline{p}$.

In order to give a subdirect representation of the $C$-set $(S_\perp, M)$, we shall consider the collection of all maximal congruences on the ada $M$ so that for each such congruence $\theta$, we have $M/\theta \cong \mathbb{3}$. We shall produce an equivalence relation $E_\theta$ on $S_\perp$ such that $(E_\theta, \theta)$ is a congruence on $(S_\perp, M)$ for each $\theta$, and the intersection of the collection of congruences $(E_\theta, \theta)$ is trivial. Thus, $(S_\perp, M)$ is a subdirect product of basic $C$-sets $(S_\perp/E_\theta, M/\theta)$.

**Definition 3.4.** For each maximal congruence $\theta$ on $M$, we define a relation on $S_\perp$ by

$$E_\theta = \{(s,t) \in S_\perp \times S_\perp : \ \beta[s,t] = \beta[t,t] \ \text{ for some } \beta \in \overline{T}^\theta\}.$$

**Lemma 3.5.** *The relation* $E_\theta$ *is an equivalence on* $S_\perp$*.*

*Proof.* Since $T[s,s] = T[s,s]$ and $T \in \overline{T}^\theta$, we have $(s,s) \in E_\theta$ so that the binary relation $E_\theta$ on $S_\perp$ is reflexive.

For symmetry, let $(s,t) \in E_\theta$. Then there exists $\beta \in \overline{T}^\theta$ such that $\beta[s,t] = \beta[t,t]$. Using (26), we have $\beta[t,s] = \beta[\beta[t,t], s] = \beta[\beta[s,t], s] = \beta[s,s]$ so that $(t,s) \in E_\theta$.

Let $(s,t),(t,r) \in E_\theta$. Then there exist $\alpha, \beta \in \overline{T}^\theta$, such that $\alpha[s,t] = \alpha[t,t]$ and $\beta[t,r] = \beta[r,r]$. As $\theta$ is a congruence on $M$, $(\alpha, T), (\beta, T) \in \theta$ implies $(\alpha \wedge \beta, T) \in \theta$ so that $(\alpha \wedge \beta) \in \overline{T}^\theta$. Note that

$$\begin{array}{rcll} (\alpha\wedge\beta)[s,r] & = & (\alpha\wedge\beta)[(\alpha\wedge\beta)[s,t],r] & \text{from (26)} \\ & = & (\alpha\wedge\beta)[(\alpha\wedge\beta)[t,t],r] & \text{from } \alpha[s,t]=\alpha[t,t] \text{ and (30)} \\ & = & (\alpha\wedge\beta)[t,r] & \text{from (26)} \\ & = & (\alpha\wedge\beta)[r,r] & \text{from } \beta[t,r]=\beta[r,r] \text{ and Proposition 2.8(v)}. \end{array}$$

Hence $(s,r) \in E_\theta$ so that $E_\theta$ is transitive. □

*Remark* 3.6. Note that, as $\theta$ is a maximal congruence on the ada $M$, $M/\theta$ must be simple, i.e., $M/\theta \cong \mathbb{3}$. Further, the quotient set $S_\perp/E_\theta$ can be treated as a pointed set with base point $\overline{\perp}$. Thus $(S_\perp/E_\theta, M/\theta)$ is a basic $C$-set under the action

$$\overline{\alpha}^\theta[\overline{s}^{E_\theta}, \overline{t}^{E_\theta}] = \begin{cases} \overline{s}^{E_\theta}, & \text{if } \alpha \in \overline{T}^\theta; \\ \overline{t}^{E_\theta}, & \text{if } \alpha \in \overline{F}^\theta; \\ \overline{\perp}^{E_\theta}, & \text{if } \alpha \in \overline{U}^\theta. \end{cases}$$

**Proposition 3.7.** *For any $\alpha \in M$, $\beta = \neg(\alpha^\downarrow \vee (\neg\alpha)^\downarrow) \vee U$ satisfies $\beta \wedge \alpha = U$. Moreover, if $(\alpha, U) \in \theta$ then $(\beta, T) \in \theta$.*

*Proof.* Since $\mathbb{3}$ is the only subdirectly irreducible ada, it is sufficient to check the validity of the identity $\beta \wedge \alpha = U$ in $\mathbb{3}$.

If $\alpha = T$, then $\beta = \neg(T^\downarrow \vee F^\downarrow) \vee U = \neg(T \vee F) \vee U = F \vee U = U$.
If $\alpha = F$, then $\beta = \neg(F^\downarrow \vee T^\downarrow) \vee U = \neg(F \vee T) \vee U = F \vee U = U$.
If $\alpha = U$, then $\beta = \neg(U^\downarrow \vee U^\downarrow) \vee U = \neg(F \vee F) \vee U = T \vee U = T$.

In all these three cases, it is straightforward to see that $\beta \wedge \alpha = U$.

Suppose $(\alpha, U) \in \theta$. Since $\theta$ is a congruence, we have $(\alpha^\downarrow, U^\downarrow) = (\alpha^\downarrow, F) \in \theta$. Also, we have $(\neg\alpha, \neg U) = (\neg\alpha, U) \in \theta$ so that $((\neg\alpha)^\downarrow, F) \in \theta$. Now, by substitution with respect to $\vee$, we have $(\alpha^\downarrow \vee (\neg\alpha)^\downarrow, F) \in \theta$.

This further implies $(\neg(\alpha^\downarrow \vee (\neg\alpha)^\downarrow), \neg F) = (\neg(\alpha^\downarrow \vee (\neg\alpha)^\downarrow), T) \in \theta$. However, since $(U,U) \in \theta$, we have $(\neg(\alpha^\downarrow \vee (\neg\alpha)^\downarrow) \vee U, T \vee U) = (\neg(\alpha^\downarrow \vee (\neg\alpha)^\downarrow) \vee U, T) \in \theta$. Hence $(\beta, T) \in \theta$. □

**Proposition 3.8.** *For each $\alpha \in M$ and each $s,t \in S_\perp$, we have the following:*

(i) $(\alpha, T) \in \theta \Rightarrow (\alpha[s,t], s) \in E_\theta$.
(ii) $(\alpha, F) \in \theta \Rightarrow (\alpha[s,t], t) \in E_\theta$.
(iii) $(\alpha, U) \in \theta \Rightarrow (\alpha[s,t], \perp) \in E_\theta$.

*Proof.*

(i) From (26), we have $\alpha[\alpha[s,t],s] = \alpha[s,s]$. Hence $(\alpha[s,t],s) \in E_\theta$ as $\alpha \in \overline{T}^\theta$.
(ii) Note that $(\alpha, F) \in \theta$ implies $(\neg\alpha, T) \in \theta$. Using (27) and (25), $(\neg\alpha)[\alpha[s,t],t] = \alpha[t, \alpha[s,t]] = \alpha[t,t] = (\neg\alpha)[t,t]$. Thus $(\alpha[s,t],t) \in E_\theta$.
(iii) If $(\alpha, U) \in \theta$, then by Proposition 3.7, $\beta = \neg(\alpha^\downarrow \vee (\neg\alpha)^\downarrow) \vee U \in \overline{T}^\theta$, and $\beta \wedge \alpha = U$. Note that

$$\begin{aligned}
\beta[\alpha[s,t],t] &= (\beta \wedge \alpha)[s,t] && \text{from (28)}\\
&= U[s,t] && \text{from Proposition 3.7}\\
&= \bot && \text{from (23)}\\
&= \beta[\bot,\bot] && \text{from Proposition 2.8(i).}
\end{aligned}$$

Consequently, by Proposition 2.8(iii), we have $\beta[\alpha[s,t],\bot] = \beta[\bot,\bot]$. Hence $(\alpha[s,t],\bot) \in E_\theta$.

□

**Lemma 3.9.** *The pair $(E_\theta, \theta)$ is a $C$-set congruence.*

*Proof.* In view of Remark 3.6, $(S_\bot/E_\theta, M/\theta)$ is a basic $C$-set. Consider the canonical maps $\nu_1 : S_\bot \to S_\bot/E_\theta$, given by $\nu_1(s) = \overline{s}^{E_\theta}$, and $\nu_2 : M \to M/\theta \cong \mathfrak{Z}$, given by $\nu_2(\alpha) = \overline{\alpha}^\theta$. We show that the pair $(\nu_1, \nu_2)$ is a $C$-set homomorphism so that $\ker(\nu_1, \nu_2) = (E_\theta, \theta)$ is a $C$-set congruence.

It is straightforward to see that $\nu_1(\bot) = \overline{\bot}^{E_\theta}$ and thus $\nu_1$ is a homomorphism of pointed sets. It is also clear that $\nu_2$ is a homomorphism of adas. Additionally, we require that $\nu_1(\alpha[s,t]) = (\nu_2(\alpha))[\nu_1(s), \nu_1(t)]$. In order to prove this, it suffices to consider the following three cases in view of the maximality of congruence $\theta$.

*Case I:* If $\alpha \in \overline{T}^\theta$, then we effectively need to show that $\overline{\alpha[s,t]}^{E_\theta} = \overline{\alpha}^\theta[\overline{s}^{E_\theta}, \overline{t}^{E_\theta}]$. From Remark 3.6 and the fact that $\alpha \in \overline{T}^\theta$, we have $\overline{\alpha}^\theta[\overline{s}^{E_\theta}, \overline{t}^{E_\theta}] = \overline{s}^{E_\theta}$. This reduces to showing that $(\alpha[s,t], s) \in E_\theta$, which follows from Proposition 3.8(i).

*Case II:* In a similar vein, if $\alpha \in \overline{F}^\theta$, we need to show that $(\alpha[s,t], t) \in E_\theta$, which follows from Proposition 3.8(ii).

*Case III:* Similarly, if $\alpha \in \overline{U}^\theta$, we require that $(\alpha[s,t], \bot) \in E_\theta$, which is precisely Proposition 3.8(iii).

This completes the proof. □

**Lemma 3.10.** *For the $C$-set $(M, M)$ the equivalence $E_\theta$ on $M$, denoted by $E_{\theta_M}$, is a subset of $\theta$.*

*Proof.* Let $(\alpha, \beta) \in E_{\theta_M}$. Then there exists $\gamma \in \overline{T}^\theta$ such that $\gamma[\![\alpha, \beta]\!] = \gamma[\![\beta, \beta]\!]$. In other words,

$$(\gamma \wedge \alpha) \vee (\neg\gamma \wedge \beta) = (\gamma \wedge \beta) \vee (\neg\gamma \wedge \beta) \tag{33}$$

Since $\gamma \in \overline{T}^\theta$, we have $(\gamma, T) \in \theta$. Moreover, $(\alpha, \alpha) \in \theta$ as $\theta$ is reflexive. It follows that $(\gamma \wedge \alpha, T \wedge \alpha) = (\gamma \wedge \alpha, \alpha) \in \theta$. Similarly, $(\gamma, T) \in \theta$ implies that $(\neg\gamma, F) \in \theta$, and as $(\beta, \beta) \in \theta$, we have $(\neg\gamma \wedge \beta, F \wedge \beta) = (\neg\gamma \wedge \beta, F) \in \theta$. Consequently $((\gamma \wedge \alpha) \vee (\neg\gamma \wedge \beta), \alpha \vee F) = ((\gamma \wedge \alpha) \vee (\neg\gamma \wedge \beta), \alpha) \in \theta$. Following a similar procedure, using $(\gamma, T), (\neg\gamma, F), (\beta, \beta) \in \theta$, we obtain $((\gamma \wedge \beta) \vee (\neg\gamma \wedge \beta), \beta) \in \theta$. Now using (33), the symmetry and transitivity of $\theta$, we have $(\alpha, \beta) \in \theta$ so that $E_{\theta_M} \subseteq \theta$. □

**Lemma 3.11.** $\bigcap_\theta E_\theta = \Delta_{S_\bot}$, *where $\theta$ ranges over all maximal congruences on $M$.*

*Proof.* By Corollary 2.11, $(S_\perp, M_\#)$ is a $B$-set so that $(S_\perp, M_\#)$ is a subdirect product of basic $B$-sets (cf. Theorem 1.6). Hence, $(S_\perp, M_\#)$ is a subalgebra of a product of basic $B$-sets $(S_x, \mathbb{2})$, where $x$ ranges over some set $X$. That is,

$$\begin{aligned}(S_\perp, M_\#) &\leq \prod_{x\in X}(S_x, \mathbb{2}) \\ &= \left(\prod_{x\in X} S_x, \mathbb{2}^X\right) \\ &\leq \left(\Big(\bigcup_{x\in X} S_x\Big)^X, \mathbb{2}^X\right)\end{aligned}$$

Note that the action in both $\left(\prod_{x\in X} S_x, \mathbb{2}^X\right)$ and $\left((\bigcup_{x\in X} S_x)^X, \mathbb{2}^X\right)$ is

$$(\alpha[s,t])(x) = \begin{cases} s(x), & \text{if } \alpha(x) = T; \\ t(x), & \text{if } \alpha(x) = F. \end{cases}$$

Also note that the action in $\left(\prod_{x\in X} S_x, \mathbb{2}^X\right)$ is simply a restriction of that on $\left((\bigcup_{x\in X} S_x)^X, \mathbb{2}^X\right)$. Since $(S_\perp, M_\#)$ is a subalgebra of $\left((\bigcup_{x\in X} S_x)^X, \mathbb{2}^X\right)$, we can see that $M_\#$ is a subalgebra of $\mathbb{2}^X$. Using the construction mentioned in Remark 1.21, $M \cong (M_\#)^\star \leq \mathbb{3}^X$.

Now for any $x_o \in X$, treating $M$ as a subalgebra of $\mathbb{3}^X$ we define maximal congruences on $M$ as follows.

$$(\alpha, \beta) \in \theta_{x_o} \Leftrightarrow \alpha(x_o) = \beta(x_o).$$

Such $\theta_{x_o}$ is indeed a maximal congruence on $M$. It is clearly an equivalence relation on $M$. Let $(\alpha_1, \beta_1), (\alpha_2, \beta_2) \in \theta_{x_o}$. Then $\alpha_1(x_o) = \beta_1(x_o)$ and $\alpha_2(x_o) = \beta_2(x_o)$. Thus $\alpha_1(x_o) \wedge \alpha_2(x_o) = \beta_1(x_o) \wedge \beta_2(x_o)$. Thus $(\alpha_1 \wedge \alpha_2, \beta_1 \wedge \beta_2) \in \theta_{x_o}$. Similarly $\theta_{x_o}$ is compatible with the other operations on $M$, viz., $\vee, \neg$ and $^\downarrow$. Note that $M$ has only the following three equivalence classes with respect to $\theta_{x_o}$.

$$\begin{aligned}\overline{\mathbf{T}}^\theta &= \{\alpha \in M : \alpha(x_o) = T\} \\ \overline{\mathbf{F}}^\theta &= \{\alpha \in M : \alpha(x_o) = F\} \\ \overline{\mathbf{U}}^\theta &= \{\alpha \in M : \alpha(x_o) = U\}\end{aligned}$$

Thus $M/\theta_{x_o}$ is simple and so $\theta_{x_o}$ is maximal.

We now show that $\bigcap E_\theta = \Delta_{S_\perp}$. Let $(s,t) \in \bigcap E_\theta$. Then for every maximal congruence $\theta$ on $M$, there exists a $\beta_\theta \in \overline{T}^\theta$ such that $\beta_\theta[s,t] = \beta_\theta[t,t]$. On using Proposition 3.1 we have $\beta_\theta^\downarrow[s,t] = \beta_\theta^\downarrow[t,t]$. Note that if $\beta_\theta$ is in $\overline{T}^\theta$, so is $\beta_\theta^\downarrow$. Moreover, $\beta_\theta^\downarrow \in M_\#$.

As $S_\perp \leq (\bigcup_{x\in X} S_x)^X$, we may treat $s, t$ as functions $s', t' \in (\bigcup_{x\in X} S_x)^X$. Note that the `if-then-else` action in $(S_\perp, M_\#)$ can be treated as a restriction of that on $((\bigcup_{x\in X} S_x)^X, \mathbb{2}^X)$. Considering the maximal congruences defined above, for each $x_o \in X$ there exists $\beta_{\theta_{x_o}}^\downarrow \in \overline{T}^{\theta_{x_o}}$, that is, $\beta_{\theta_{x_o}}^\downarrow(x_o) = T$ and $\beta_{\theta_{x_o}}^\downarrow[s', t'] = \beta_{\theta_{x_o}}^\downarrow[t', t']$. In other words, for each $x \in X$, $(\beta_{\theta_{x_o}}^\downarrow[s', t'])(x) = (\beta_{\theta_{x_o}}^\downarrow[t', t'])(x)$. In particular for $x = x_o$, $(\beta_{\theta_{x_o}}^\downarrow[s', t'])(x_o) = (\beta_{\theta_{x_o}}^\downarrow[t', t'])(x_o)$.

However $(\beta^{\downarrow}_{\theta_{x_o}}[s',t'])(x_o) = s'(x_o)$ as $\beta^{\downarrow}_{\theta_{x_o}}(x_o) = T$. Similarly, $(\beta^{\downarrow}_{\theta_{x_o}}[t',t'])(x_o) = t'(x_o)$.

This tells us that for each $x_o \in X$, $s'(x_o) = t'(x_o)$, that is, $s' \equiv t'$ which means that $s = t$ in $S_\perp$. This completes the proof. □

*Remark* 3.12. Let $\alpha, \beta \in M$ with $\alpha \neq \beta$. Treating $M$ as a subalgebra of $\mathtt{3}^X$ for some $X$, there exists $x_o \in X$ such that $\alpha(x_o) \neq \beta(x_o)$. Then $\theta_{x_o}$, as in the previous proof, is a maximal congruence which clearly separates $\alpha$ and $\beta$. Since the intersection of all such congruences is $\Delta_M$, the intersection of all maximal congruences on $M$

$$\bigcap_{\theta \text{ maximal}} \theta = \Delta_M$$

We now prove the main theorem of this section.

**Theorem 3.13.** *Every $C$-set $(S_\perp, M)$ where $M$ is an ada is a subdirect product of basic $C$-sets.*

*Proof.* Let $(S_\perp, M)$ be a $C$-set where $M$ is an ada and $\{\theta\}$ be the collection of all maximal congruences on $M$. By Lemma 3.9, for each $\theta$, the pair $(E_\theta, \theta)$ is a $C$-set congruence on $(S_\perp, M)$ and by Remark 3.6 $(S_\perp/E_\theta, M/\theta)$ is a basic $C$-set. Further, by Lemma 3.11 and Remark 3.12, the intersection of all congruences $(E_\theta, \theta)$ is trivial. Hence, $(S_\perp, M)$ is a subdirect product of $(S_\perp/E_\theta, \mathtt{3})$, where $\theta$ varies over maximal congruences on $M$. □

The following consequence of Theorem 3.13 is useful to establish the equivalence between programs which admit the current setup.

**Corollary 3.14.** *An identity (quasi-identity) is satisfied in every $C$-set $(S_\perp, M)$ where $M$ is an ada if and only if it is satisfied in all basic $C$-sets.*

## 4. Agreeable $C$-sets

In this section, we describe an algebraic formalism for the equality test over possibly non-halting programs. The equality test over the functions $f, g \in \mathcal{T}_o(X_\perp)$ can be naturally described by the following:

$$(f * g)(x) = \begin{cases} T, & \text{if } f(x) = g(x) \text{ and } f(x) \neq \perp \neq g(x); \\ F, & \text{if } f(x) \neq g(x) \text{ and } f(x) \neq \perp \neq g(x); \\ U, & \text{otherwise.} \end{cases} \tag{34}$$

For simplicity of notation, we will denote the condition $f(x) = g(x)$ and $f(x) \neq \perp \neq g(x)$ by $f(x) = g(x)$ $(\neq \perp)$ and the condition $f(x) \neq g(x)$ and $f(x) \neq \perp \neq g(x)$ by $f(x) \neq g(x)$ $(\neq \perp)$. Consequently, $f * g$ can be identified with the pair of sets $(A, B)$ on $X$, where $A = \{x \in X : f(x) = g(x)\ (\neq \perp)\}$ and $B = \{x \in X : f(x) \neq g(x)\ (\neq \perp)\}$.

Keeping this model in mind, we extend the notion of agreeable $B$-sets, given by Jackson and Stokes in [11], and define agreeable $C$-sets as follows.

**Definition 4.1.** A $C$-set $(S_\perp, M)$ equipped with a function

$$* : S_\perp \times S_\perp \to M$$

is said to be *agreeable* if it satisfies the following axioms for all $s, t, u, v \in S_\perp$ and $\alpha \in M$:

$$(s * s)[s, \perp] = s \qquad \text{(domain axiom)} \tag{35}$$

$$\perp * s = U = s * \perp \qquad \text{($\perp$-comparison)} \tag{36}$$

$$(s * t)[s, t] = (s * t)[t, t] \qquad \text{(equality on conclusions)} \tag{37}$$

$$\alpha[s, t] * \alpha[u, v] = \alpha[\![s * u, t * v]\!] \qquad \text{(operation interchange)} \tag{38}$$

$$((s * s = T) \wedge (s * t = U)) \Rightarrow t = \perp \qquad \text{(totality condition)} \tag{39}$$

While the operation interchange axiom (38) is indeed an axiom in the context of agreeable $B$-sets (cf. axiom (9)), one can verify that axiom (37) holds in agreeable $B$-sets. However, the other axioms are specific to the current scenario of the non-halting case. These axioms can be justified along the following lines by considering equality of functions over the functional model $(\mathcal{T}_o(X_\perp), 3^X)$ of $C$-sets.

In $\mathcal{T}_o(X_\perp)$, the domain of a function is considered in the spirit of a partial function, i.e., all those points whose image is not $\perp$. In the model $(\mathcal{T}_o(X_\perp), 3^X)$, the partial predicate $s * s$ represents the domain of $s$. The domain axiom (35) captures the behaviour of `if-then-else` with respect to the domain of $s$. For instance, we expect $s * s$ takes truth value $T$ in the domain of $s$ so that $(s * s)[s, \perp] = s$. Also, in the complement of the domain of $s$, $s * s$ should take value $U$ so that $(s * s)[s, \perp] = U[s, \perp] = \perp = s$.

Note that we check the equality of two functions over their domains. Thus the $\perp$-comparison axiom (36) states that comparing the `error` state $\perp$ with any element $s$ results in the `undefined` predicate $U$.

The axiom of equality on conclusions (37) exhibits the behaviour of equality test $*$ on conclusions of the `if-then-else` action of the $C$-set. Indeed, when the partial predicate $s * t = T$, $(s * t)[s, t] = s = t = (s * t)[t, t]$ and similarly if $s * t = F$, then $(s * t)[s, t] = t = (s * t)[t, t]$. Further, if $s * t = U$, then $(s * t)[s, t] = \perp = (s * t)[t, t]$.

The axiom of operation interchange (38) describes how $*$ and the `if-then-else` action relate to the action on the $C$-set $(M, M)$. The totality condition (39) is a quasi-identity, in which if $s$ is a total function but $s * t$ is undefined, then it must follow that $t$ is the empty function, i.e., $t = \zeta_\perp$.

Thus we arrive at the following example of agreeable $C$-sets.

**Example 4.2.** The pair $(\mathcal{T}_o(X_\perp), 3^X)$ is an agreeable $C$-set under the operation $*$ defined in (34). For verification of the axioms (35) – (39) refer to Appendix A.3. Such agreeable $C$-sets are called *agreeable functional $C$-sets.*

**Example 4.3.** Every basic $C$-set is agreeable under the operation given by

$$s * t = \begin{cases} T, & \text{if } s = t \ (\neq \perp); \\ F, & \text{if } s \neq t \ (\neq \perp); \\ U, & \text{if } s = \perp \text{ or } t = \perp. \end{cases} \tag{40}$$

One can easily verify that the axioms (35) – (39) hold here. Such agreeable $C$-sets will be called *agreeable basic $C$-sets.*

**Proposition 4.4.** *The operation defined in* (40) *is the only possible operation under which a basic $C$-set can be made agreeable.*

*Proof.* We shall show that for a basic $C$-set, axioms (35) to (39) restrict the operation $*$ to precisely (40). Let $(S_\perp, 3)$ be a basic $C$-set which is agreeable, that is, it is equipped with an operation $*: S_\perp \times S_\perp \to \mathbb{3}$ which satisfies (35) - (39). Consider the following cases:

(i) *Case I*: $s = \perp$ or $t = \perp$: Then from (36), we have $s * t = U$.

(ii) *Case II*: $s = t$ $(\neq \perp)$: We will show that neither $s * t = F$ nor $s * t = U$ is possible. Consequently, it must be the case that $s * t = T$.

Assume that $s * t = F$. This, in conjunction with the hypothesis $s = t$ and (35), gives that $\perp = F[s, \perp] = (s * t)[s, \perp] = (s * s)[s, \perp] = s$, a contradiction to our assumption that $s \neq \perp$. If $s * t = U$, along similar lines, we obtain $\perp = U[s, \perp] = (s * s)[s, \perp] = s$, a contradiction.

(iii) *Case III*: $s \neq t$ $(\neq \perp)$: Along similar lines, we will show that $s * t \notin \{T, U\}$, which would imply that $s * t = F$.

Assume that $s * t = T$. It follows from (37) that $s = T[s, t] = (s * t)[s, t] = (s*t)[t, t] = T[t, t] = t$, a contradiction to the hypothesis $s \neq t$. Note that if $S_\perp$ has exactly two distinct elements then this case would be redundant. Suppose that $s*t = U$. *Case II* proved above, in conjunction with the hypothesis that $s \neq \perp$, gives that $s * s = T$. From the statements $s * s = T$, $s * t = U$ and quasi-identity (39), we have $t = \perp$, a contradiction.

Thus the operation $*$ must be as defined in (40). $\square$

**Example 4.5.** The $C$-set $(M, M)$ is agreeable under the operation

$$\alpha * \beta = (\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta).$$

The operation can be equivalently expressed in terms of the `if-then-else` action by

$$\alpha * \beta = \alpha[\![\beta, \neg\beta]\!].$$

Along similar lines as Example 2.3, the axioms (35) – (39) can be verified in the $C$-algebra $\mathbb{3}$ to observe that $(M, M)$ is an agreeable $C$-set.

*Remark* 4.6. If the $C$-algebra $M$ is $\mathbb{3}^X$, the equality test on the agreeable $C$-set $(M, M)$ coincides with that of the functional case, as shown below:

$$(\alpha * \beta)(x) = \begin{cases} T, & \text{if } \alpha(x) = \beta(x)\ (\neq U); \\ F, & \text{if } \alpha(x) \neq \beta(x)\ (\neq U); \\ U, & \text{otherwise.} \end{cases}$$

We now prove a representation theorem of agreeable $C$-sets along the lines of Theorem 3.13.

**Theorem 4.7.** *Every agreeable $C$-set $(S_\perp, M)$ where $M$ is an ada is a subdirect product of agreeable basic $C$-sets.*

*Proof.* Let $(S_\perp, M)$ be an agreeable $C$-set where $M$ is an ada. For every maximal congruence $\theta$ on $M$, consider the pair $(E_\theta, \theta)$ as in Definition 3.4. By Lemma 3.9, we have already ascertained that for each $\theta$, the pair $(E_\theta, \theta)$ is a $C$-set congruence on $(S_\perp, M)$ and by Remark 3.6 that $(S_\perp/E_\theta, M/\theta)$ is a basic $C$-set. In order to ascertain that this pair is indeed a congruence in the context of agreeable $C$-sets, it is sufficient to show that

$$(a_1, a_2), (b_1, b_2) \in E_\theta \Rightarrow (a_1 * b_1, a_2 * b_2) \in \theta.$$

Let $(a_1, a_2), (b_1, b_2) \in E_\theta$. Then there exist $\alpha$ and $\beta \in \overline{T}^\theta$ such that

$$\alpha[a_1, a_2] = \alpha[a_2, a_2] \tag{41}$$

$$\beta[b_1, b_2] = \beta[b_2, b_2] \tag{42}$$

Note that $(\alpha, T), (\beta, T) \in \theta$ implies that $(\alpha \wedge \beta, T \wedge T) = (\alpha \wedge \beta, T) \in \theta$. Applying (30) on (41) and Proposition 2.8(v) on (42), we have, for $(\alpha \wedge \beta) \in \overline{T}^\theta$

$$(\alpha \wedge \beta)[a_1, a_2] = (\alpha \wedge \beta)[a_2, a_2] \tag{43}$$

$$(\alpha \wedge \beta)[b_1, b_2] = (\alpha \wedge \beta)[b_2, b_2] \tag{44}$$

These imply that

$$(\alpha \wedge \beta)[a_1, a_2] * (\alpha \wedge \beta)[b_1, b_2] = (\alpha \wedge \beta)[a_2, a_2] * (\alpha \wedge \beta)[b_2, b_2]. \tag{45}$$

From (38) it follows that

$$(\alpha \wedge \beta)[\![a_1 * b_1, a_2 * b_2]\!] = (\alpha \wedge \beta)[\![a_2 * b_2, a_2 * b_2]\!],$$

so that $(a_1 * b_1, a_2 * b_2) \in E_{\theta_M} \subseteq \theta$, by Lemma 3.10. Further, by Lemma 3.11 and Remark 3.12, the intersection of all congruences $(E_\theta, \theta)$, where $\theta$ ranges over all maximal congruences of $M$, is trivial. This completes the proof. □

**Corollary 4.8.** *An identity (quasi-identity) is satisfied in every agreeable $C$-set $(S_\perp, M)$ where $M$ is an ada if and only if it is satisfied in all agreeable basic $C$-sets.*

In view of Corollary 4.8 and (40), we have the following result.

**Corollary 4.9.** *In every agreeable $C$-set $(S_\perp, M)$ where $M$ is an ada we have $s * t = t * s$.*

Note that the only axiom of agreeable $C$-sets that plays a role in the proof of Theorem 4.7 is (38). The remaining axioms have been included in order that the operation on agreeable basic $C$-sets be uniquely defined. The proof of Theorem 4.7 suggests an alternative proof for Theorem 1.10, without using the commutativity of $*$, which we now present.

**Theorem 4.10** ([11])**.** *Every agreeable $B$-set $(S, B)$ is a subdirect product of basic agreeable $B$-sets.*

*Proof.* Let $F$ be an ultrafilter of $B$. Consider the relation $E_F = \{(s, t) \in S \times S : \gamma[s, t] = t \text{ for some } \gamma \in F\}$ as defined in [11]. The pair $(E_F, F)$ is a $B$-set congruence. In order to show that the pair $(E_F, F)$ is a congruence on agreeable $B$-sets, we show that

$$(a_1, a_2), (b_1, b_2) \in E_F \Rightarrow (a_1 * b_1, a_2 * b_2) \in \theta_F,$$

where $\theta_F$ is the congruence on $B$ induced by the ultrafilter $F$.

Since $(a_1, a_2), (b_1, b_2) \in E_F$, there exist $\alpha, \beta \in F$ such that

$$\alpha[a_1, a_2] = a_2 \tag{46}$$

$$\beta[b_1, b_2] = b_2 \tag{47}$$

In view of the commutativity of $\wedge$, (6), (46) and (1), we obtain $(\alpha \wedge \beta)[a_1, a_2] = (\beta \wedge \alpha)[a_1, a_2] = \beta[\alpha[a_1, a_2], a_2] = \beta[a_2, a_2] = a_2$. Similarly we can obtain $(\alpha \wedge \beta)[b_1, b_2] = b_2$. This implies that

$$(\alpha \wedge \beta)[a_1, a_2] * (\alpha \wedge \beta)[b_1, b_2] = a_2 * b_2$$

From axiom (9), we can deduce that

$$(\alpha \wedge \beta)[a_1 * b_1, a_2 * b_2] = a_2 * b_2.$$

Since $F$ is an ultrafilter of $B$, it suffices to ascertain that

$$a_1 * b_1 \in F \Leftrightarrow a_2 * b_2 \in F.$$

Assume that $a_1 * b_1 \in F$. Since $\alpha \wedge \beta \in F$, we have $(\alpha \wedge \beta) \wedge (a_1 * b_1) \in F$. Further, as $F$ is a filter and $(\alpha \wedge \beta) \wedge (a_1 * b_1) \leq ((\alpha \wedge \beta) \wedge (a_1 * b_1)) \vee (\neg(\alpha \wedge \beta) \wedge (a_2 * b_2)) = (\alpha \wedge \beta)[a_1 * b_1, a_2 * b_2]$ we have $(\alpha \wedge \beta)[a_1 * b_1, a_2 * b_2] = a_2 * b_2 \in F$.

Conversely, assume that $a_2 * b_2 \in F$. The symmetry of equivalence relation $E_F$ implies that $(a_2, a_1), (b_2, b_1) \in E_F$. Along similar lines as above, we obtain $a_1 * b_1 \in F$. This completes the proof. □

## 5. Conclusion

The axiomatization of systems based on various program constructs are extremely useful in the study of program semantics in general, and in establishing program equivalence in particular. While many authors have studied the axiomatization of the `if-then-else` construct, the current work considered the case where the programs and tests could possibly be non-halting. In this connection, this work introduced the notion of $C$-sets to axiomatize the systems of `if-then-else` in which the tests are drawn from an abstract $C$-algebra. The axioms of $C$-sets include a quasi-identity for $\wedge$-compatibility along with various other identities. When the $C$-algebra is an ada, we obtained a subdirect representation of $C$-sets through basic $C$-sets. This in turn establishes the completeness of the axiomatization and paves the way for determining the equivalence of programs under consideration through basic $C$-sets. Further, in order to axiomatize `if-then-else` systems with the equality test, this work extended the concept of $C$-sets to agreeable $C$-sets and obtained similar results.

As future work, one may investigate a complete axiomatization of the systems under consideration in which all the axioms are equations. It is also desirable to extend the results to the general case of $C$-sets without restricting the $C$-algebra to be an ada. Another natural extension of the current work is to investigate the axiomatization of systems that include the composition of programs. We address this question by considering a semigroup structure on the program sort of $C$-sets in one of our forthcoming papers.

## Acknowledgements

We are thankful to the referee for providing insightful comments, which have improved the presentation of the paper.

## References


[1] N. D. Belnap. Conditional assertion and restricted quantification. *Noûs*, pages 1–12, 1970.

[2] G. M. Bergman. Actions of Boolean rings on sets. *Algebra Universalis*, 28:153–187, 1991.

[3] J. Bergstra, I. Bethke, and P. Rodenburg. A propositional logic with 4 values: true, false, divergent and meaningless. *Journal of Applied Non-Classical Logics*, 5:199–217, 1995.

[4] S. L. Bloom and R. Tindell. Varieties of "if-then-else". *SIAM J. Comput.*, 12:677–707, 1983.

[5] D. A. Bochvar. Ob odnom tréhznacnom isčislenii i égo priménénii k analiza paradoksov klassičéskogo răssirénnogo funkcional'nogo isčisléniá (in Russian). *Matématičeskij Sbornik*, 4(46): 287–308, 1938. Translated to English by M. Bergmann "On a three-valued logical calculus and its application to the analysis of the paradoxes of the classical extended functional calculus". *History and Philosophy of Logic*, 2:87–112, 1981.

[6] J. W. de Bakker. Semantics of programming languages. In *Advances in Information Systems Science*, pages 173–227. Springer, 1969.

[7] I. Guessarian and J. Meseguer. On the axiomatization of "if-then-else". *SIAM J. Comput.*, 16:332–357, 1987.

[8] F. Guzmán and C. C. Squier. The algebra of conditional logic. *Algebra Universalis*, 27:88–110, 1990.

[9] A. Heyting. Die formalen regeln der intuitionistischen logik, sitzungsberichte der preuszischen akademie der wissenschaften, physikalischmathematische klasse, (1930), 42–56 57–71 158–169 in three parts. *Sitzungsber. preuss. Akad. Wiss*, 42:158–169, 1934.

[10] S. Igarashi. Semantics of ALGOL-like statements. In *Symposium on Semantics of Algorithmic Languages*, pages 117–177. Springer, 1971.

[11] M. Jackson and T. Stokes. Semigroups with if-then-else and halting programs. *Int. J. Algebra Comput.*, 19:937–961, 2009.

[12] M. Jackson and T. Stokes. Monoids with tests and the algebra of possibly non-halting programs. *J. Log. Algebr. Methods Program.*, 84:259–275, 2015.

[13] J. F. Kennison. Triples and compact sheaf representation. *J. Pure Appl. Algebra*, 20:13–38, 1981.

[14] S. Kleene. On notation for ordinal numbers. *The Journal of Symbolic Logic*, 3:150–155, 1938.

[15] S. C. Kleene. *Introduction to metamathematics*. D. Van Nostrand Co., Inc., New York, N. Y., 1952.

[16] J. Lukasiewicz. On three-valued logic. Ruch Filozoficzny, 5,(1920), English translation in Borkowski, L.(ed.) 1970. Jan Lukasiewicz: Selected Works, 1920.

[17] E. Manes. Adas and the equational theory of if-then-else. *Algebra Universalis*, 30:373–394, 1993.

[18] E. G. Manes. A transformational characterization of if-then-else. *Theoretical Computer Science*, 71:413–417, 1990.

[19] J. McCarthy. A basis for a mathematical theory of computation. In *Computer programming and formal systems*, pages 33–70. North-Holland, Amsterdam, 1963.

[20] A. H. Mekler and E. M. Nelson. Equational bases for if-then-else. *SIAM J. Comput.*, 16:465–485, 1987.

[21] D. Pigozzi. Equality-test and if-then-else algebras: Axiomatization and specification. *SIAM J. Comput.*, 20:766–805, 1991.

[22] R. Sethi. Conditional expressions with equality tests. *J. ACM*, 25:667–674, 1978.

[23] T. Stokes. Comparison semigroups and algebras of transformations. *Semigroup Forum*, 81:325–334, 2010.

[24] T. Stokes. Sets with $B$-action and linear algebra. *Algebra Universalis*, 39:31–43, 1998.


## Appendix A. Proofs

### A.1. **Verification of Example 2.3.**

Let $\alpha, \beta \in M$.
Axiom (23): $U[\![\alpha, \beta]\!] = (U \wedge \alpha) \vee (\neg U \wedge \beta) = U \vee U = U$.

Axiom (24): $F[\![\alpha, \beta]\!] = (F \wedge \alpha) \vee (\neg F \wedge \beta) = F \vee \beta = \beta$.

Axiom (25): Note that $(\neg\alpha)[\![\beta,\gamma]\!] = (\neg\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$. On the other hand $\alpha[\![\gamma,\beta]\!] = (\alpha \wedge \gamma) \vee (\neg\alpha \wedge \beta)$. We check the validity of the identity $(\neg\alpha \wedge \beta) \vee (\alpha \wedge \gamma) = (\alpha \wedge \gamma) \vee (\neg\alpha \wedge \beta)$ in the three element $C$-algebra $\mathbb{3}$.

*Case I*: $\alpha = T$: $(\neg T \wedge \beta) \vee (T \wedge \gamma) = F \vee \gamma = \gamma = \gamma \vee F = (T \wedge \gamma) \vee (\neg T \wedge \beta)$.
*Case II*: $\alpha = F$: $(\neg F \wedge \beta) \vee (F \wedge \gamma) = \beta \vee F = \beta = F \vee \beta = (F \wedge \gamma) \vee (\neg F \wedge \beta)$.
*Case III*: $\alpha = U$: $(\neg U \wedge \beta) \vee (U \wedge \gamma) = U \vee U = (U \wedge \gamma) \vee (\neg U \wedge \beta)$.

In view of Theorem 1.14, the identity $(\neg\alpha \wedge \beta) \vee (\alpha \wedge \gamma) = (\alpha \wedge \gamma) \vee (\neg\alpha \wedge \beta)$ is valid in all $C$-algebras and hence the identity $(\neg\alpha)[\![\beta,\gamma]\!] = \alpha[\![\gamma,\beta]\!]$ holds in $M$.

Along similar lines, one can verify the identities (26), (27), (28), (29) and the quasi-identity (30) in the $C$-algebra $\mathbb{3}$ by considering $\alpha$ to be $T$, $F$ and $U$ casewise. Consequently, the axioms hold in $M$.

A.2. **Verification of Example 2.4.**

In order to verify the axioms we will rely on the pairs of sets representation of the $C$-algebra $\mathbb{3}^X$ by Guzmán and Squier in [8]. Every $\alpha \in \mathbb{3}^X$ can be represented by the pair of sets $(A,B) = (\alpha^{-1}(T), \alpha^{-1}(F))$. In this representation $\mathbf{T} = (X, \emptyset)$, $\mathbf{F} = (\emptyset, X)$ and $\mathbf{U} = (\emptyset, \emptyset)$.
Let $f, g, h, k \in \mathcal{T}_o(X_\perp)$. Let $\alpha, \beta \in \mathbb{3}^X$ be represented by the pairs of sets $(A,B)$ and $(C,D)$ respectively. Note that

$$\alpha[f,g](x) = (A,B)[f,g](x) = \begin{cases} f(x), & \text{if } x \in A; \\ g(x), & \text{if } x \in B; \\ \perp, & \text{otherwise.} \end{cases}$$

Using this action one can verify the axioms (23), (24) and (25) easily.

Axiom (26): For $x \in A$ we have $(A,B)[f,g](x) = f(x)$. Consequently

$$\begin{aligned}(A,B)[(A,B)[f,g],h](x) &= \begin{cases} f(x), & \text{if } x \in A; \\ h(x), & \text{if } x \in B; \\ \perp, & \text{otherwise.} \end{cases} \\ &= (A,B)[f,h](x)\end{aligned}$$

so that $\alpha[\alpha[f,g],h] = \alpha[f,h]$.

Axiom (27) can be verified along the same lines as axiom (26).

Axiom (28): Note that

$$(\alpha \wedge \beta)[f,g](x) = (A \cap C, B \cup (A \cap D))[f,g](x) = \begin{cases} f(x), & \text{if } x \in A \cap C; \\ g(x), & \text{if } x \in B \cup (A \cap D); \\ \perp, & \text{otherwise.} \end{cases}$$

On the other hand

$$\alpha[\beta[f,g],g](x) = (A,B)[(C,D)[f,g],g](x) = \begin{cases} (C,D)[f,g](x), & \text{if } x \in A; \\ g(x), & \text{if } x \in B; \\ \bot, & \text{otherwise.} \end{cases}$$
$$= \begin{cases} f(x), & \text{if } x \in A \cap C; \\ g(x), & \text{if } x \in A \cap D; \\ \bot, & \text{if } x \in A \cap (X \setminus (C \cup D)); \\ g(x), & \text{if } x \in B; \\ \bot, & \text{otherwise.} \end{cases}$$
$$= \begin{cases} f(x), & \text{if } x \in A \cap C; \\ g(x), & \text{if } x \in B \cup (A \cap D); \\ \bot, & \text{otherwise.} \end{cases}$$

Hence $(\alpha \wedge \beta)[f,g] = \alpha[\beta[f,g],g]$. Axiom (29) can be verified along similar lines.

Axiom (30): Given that $(A,B)[f,g](x) = (A,B)[g,g](x)$ for all $x \in X_\bot$ we have $f(x) = g(x)$ for all $x \in A$, in particular for all $x \in A \cap C$. Hence

$$(\alpha \wedge \beta)[f,g](x) = (A \cap C, B \cup (A \cap D))[f,g](x) = \begin{cases} f(x), & \text{if } x \in A \cap C; \\ g(x), & \text{if } x \in B \cup (A \cap D); \\ \bot, & \text{otherwise.} \end{cases}$$
$$= \begin{cases} g(x), & \text{if } x \in A \cap C; \\ g(x), & \text{if } x \in B \cup (A \cap D); \\ \bot, & \text{otherwise.} \end{cases}$$
$$= (\alpha \wedge \beta)[g,g](x).$$

Thus $(\alpha \wedge \beta)[f,g] = (\alpha \wedge \beta)[g,g]$ and so quasi-identity (30) holds.

A.3. **Verification of Example 4.2.**

For $f, g \in \mathcal{T}_o(X_\bot)$ using the pairs of sets representation of $f * g$, axioms (35), (36) and (37) can be verified easily.

Let $f, g, h, k \in \mathcal{T}_o(X_\bot)$.

Axiom (38): We will show that $\alpha[f,g] * \alpha[h,k] = \alpha[\![f * h, g * k]\!]$. Let $\alpha = (A,B)$ and $\alpha[f,g] = \mathscr{F}_1$ where

$$\mathscr{F}_1(x) = \begin{cases} f(x), & \text{if } x \in A; \\ g(x), & \text{if } x \in B; \\ \bot, & \text{otherwise.} \end{cases}$$

Also let $\alpha[h,k] = \mathscr{F}_2$ where

$$\mathscr{F}_2(x) = \begin{cases} h(x), & \text{if } x \in A; \\ k(x), & \text{if } x \in B; \\ \bot, & \text{otherwise.} \end{cases}$$

Let $\mathscr{F}_1 * \mathscr{F}_2 = (C, D)$ where $C = \{x \in X : \mathscr{F}_1(x) = \mathscr{F}_2(x)(\neq \bot)\}$ and $D = \{x \in X : \mathscr{F}_1(x) \neq \mathscr{F}_2(x)(\neq \bot)\}$. Let $f * h = (E, F)$ where $E = \{x \in X : f(x) = h(x)(\neq \bot)\}$ and $F = \{x \in X : f(x) \neq h(x)(\neq \bot)\}$. Let $g * k = (G, H)$ where $G = \{x \in X : g(x) = k(x)(\neq \bot)\}$ and $H = \{x \in X : g(x) \neq k(x)(\neq \bot)\}$. Then we have

$$\begin{aligned}
&(A,B)[\![(E,F),(G,H)]\!] \\
&\quad = \big((A,B) \wedge (E,F)\big) \vee \big(\neg((A,B)) \wedge (G,H)\big) \\
&\quad = \big((A \cap E, B \cup (A \cap F)\big) \vee \big(B \cap G, A \cup (B \cap H)\big) \\
&\quad = \Big((A \cap E) \cup \big((B \cup (A \cap F)) \cap (B \cap G)\big), \big(B \cup (A \cap F)\big) \cap \big(A \cup (B \cap H)\big)\Big)
\end{aligned}$$

In effect we need to show that

$$(C, D) = \Big((A \cap E) \cup \big((B \cup (A \cap F)) \cap (B \cap G)\big), \big(B \cup (A \cap F)\big) \cap \big(A \cup (B \cap H)\big)\Big).$$

It is straightforward to observe that $(A \cap E) \cup \big((B \cup (A \cap F)) \cap (B \cap G)\big) \subseteq C$. The reverse inclusion can be ascertained as follows. If $x \in C$ it follows that $x \in A \cup B$. Now it can be observed that if $x \in A$ then $x \in E$ and if $x \in B$ then $x \in G$. Hence $C = (A \cap E) \cup \big((B \cup (A \cap F)) \cap (B \cap G)\big)$.
Similarly one can observe that $\big(B \cup (A \cap F)\big) \cap \big(A \cup (B \cap H)\big) \subseteq D$. For the reverse inclusion if $x \in D$ then $x \in A \cup B$. It can be observed that if $x \in A$ then $x \in F$ and if $x \in B$ then $x \in H$. Hence $D = \big(B \cup (A \cap F)\big) \cap \big(A \cup (B \cap H)\big)$.

Axiom (39): Let $f * f = (X, \emptyset)$ and $f * g = (\emptyset, \emptyset)$. Since $f * f = (X, \emptyset)$ we have $f(x) \neq \bot$ for all $x \in X$. Set $A = \{x \in X : f(x) = g(x)(\neq \bot)\}$ and $B = \{x \in X : f(x) \neq g(x)(\neq \bot)\}$. Note that $A = B = \emptyset$ as $f * g = (\emptyset, \emptyset)$. If $g(y) \neq \bot$ for some $y \in X$ then it follows that $y \in A$ or $y \in B$ as $f(y) \neq \bot$. This contradicts $A = B = \emptyset$. Hence $g(x) = \bot$ for all $x \in X$ so that $((f * f = \mathbf{T}) \wedge (f * g = \mathbf{U})) \Rightarrow g = \zeta_\bot$.

Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati, India
*E-mail address*: p.gayatri@iitg.ac.in

Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati, India
*E-mail address*: kvk@iitg.ac.in

Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, India
*E-mail address*: pbhaduri@iitg.ac.in