Paraconsistent Relations as a Variant of Kleene Algebras *

Juliana Cunha
CIDMA, Dep. Mathematics, Aveiro University
Aveiro, Portugal
INESC TEC & Dep. Informatics, Minho University
Braga, Portugal
juliana.cunha@ua.pt

Alexandre Madeira
CIDMA, Dep. Mathematics, Aveiro University
Aveiro, Portugal
madeira@ua.pt

Luís S. Barbosa
INESC TEC & Dep. Informatics, Minho University
Braga, Portugal
lsb@di.uminho.pt

Kleene algebras (KA) and Kleene algebras with tests (KAT) provide an algebraic framework to capture the behavior of conventional programming constructs. This paper explores a broader understanding of these structures, in order to enable the expression of programs and tests yielding vague or inconsistent outcomes. Within this context, we introduce the concept of a paraconsistent Kleene Algebra with tests (PKAT), capable of capturing vague and contradictory computations. Finally, to establish the semantics of such a structure, we introduce two algebras, $\mathbf{Set_P}(\mathcal{T})$ and $\mathbf{Rel_P}(\mathcal{K},\mathcal{T})$, parametric on a class of twisted structures $\mathcal{K}$ and $\mathcal{T}$. We believe this sort of structures, for their huge flexibility, have an interesting application potential.

1 Introduction

In his seminal work [27], Stephen Kleene described finite deterministic automata along with a specification language: regular expressions. Kleene's paper left open the question of whether a finite, sound, and complete axiomatization of the equivalence of regular expressions existed, which would provide an algebraic framework for describing regular languages. This question has been explored by many researchers. In 1964, Redko [37] proved that no finite set of equational axioms could fully characterize the algebra of regular expressions, and in 1966, Salomaa [38] provided two complete axiomatizations of this algebra. Conway's comprehensive 1971 work [14] presented a detailed overview of results related to regular expressions and their axiomatizations. In [29] Kozen showed that Salomaa's axiomatization is non-algebraic, i.e. unsound under substitution of alphabet symbols by arbitrary regular expressions. He then presented an algebraic axiomatization: Kleene algebras (KA). Later in [30], Kozen also introduced a variant of KA called Kleene Algebras with Tests (KAT). The addition of tests to KA was prompted by the need to express conventional constructs such as conditionals and **while** loops. As a result, KAT is specifically designed for equational reasoning about these constructs. For any proposition α it is possible to form a test α? that acts as a guard: it succeeds with no side effects in states satisfying α, and fails or aborts otherwise[1].

*This work was financed by PRR - Plano de Recuperação e Resiliência under the Next Generation EU from the European Union within Project Agenda ILLIANCE C644919832-00000035 - Project n 46, as well as by National Funds through FCT, the Portuguese Foundation for Science and Technology, within the project IBEX, with reference PTDC/CCI-COM/4280/2021.

[1]Notice that throughout this paper, distinct symbols will be used for programs and propositions. Therefore, we often omit the ? symbol and simply write α to denote a test.

Submitted to:
LSFA 2024

© J. Cunha, A. Madeira & L. S. Barbosa
This work is licensed under the
Creative Commons Attribution License.

While historically prominent in automata theory and formal languages [33], KA and their variants have found applications across various domains, including relational algebra [39], program semantics and logics [34], compiler optimization [31], and algorithm design and analysis [28]. For a more comprehensive read of its applications, the reader is referred to [4].

KAT are indeed suitable to reason about imperative programs since these programs can be thought of as sequences of discrete steps, each related to an atomic transition in a standard automaton. Traditionally, these programs operate within a bivalent truth space, where assertions have Boolean outcomes. While this framework has proven to suit a huge range of computer science applications, its inherent simplicity and rigidity may fall short in capturing some intricacies present in real-world scenarios.

Actually, it is not uncommon in software engineering to face scenarios in which vague, or weakly consistent, or even contradictory information is present. Often such characteristics cannot be abstracted away or swept under the carpet. A typical example the authors are currently facing, and which forms one of our motivations for this work, concerns repositories of medical images in a particular domain, which are labelled by several medical judgements from different expert teams, which often are in partial contradiction. Another example arises in the analysis of implementations of quantum circuits in current NISQ (Noisy Intermediate-Scale Quantum) technology [35], where conflicting decoherence levels in the quantum memory have to be taken into account.

Therefore, the introduction of algebraic structures to model computations becomes necessary when the behavior of these computations does not conform to a simple bivalent outcome. Instead, it may involve weighted outcomes from a richer domain, potentially lacking consistency. In this context, vagueness captures the lack of information, while paraconsistency, a well-established designation in logic, expresses the excess of information arising from contradictory judgments.

From a technical point of view, paraconsistency refers to a property inherent to a consequence relation. A logic is said to be "paraconsistent" iff its logical consequence relation, whether semantic or proof-theoretic, does not lead to explosion [36]. In logic, the term "explosion" refers to the principle of *ex falso quodlibet*, meaning "from contradiction, anything follows". This principle is the basis for the law of non-contradiction in classical logic. It asserts that from contradictory premises, any proposition can be derived, thus leading to triviality (where anything can be proved true). Consequently, paraconsistent logics set themselves apart from classical ones by their capacity to handle inconsistent information without "exploding" into absurdity. Initially developed in Latin America during the 1950s and 1960s, notably through the influential contributions of F. Asenjo and Newton da Costa, paraconsistent logic quickly garnered attention within the logic and computer science communities. Its original focus on mathematical applications has since expanded, as evidenced by recent literature emphasizing the engineering potential of paraconsistency [3]. Relevant applications are documented in several fields, including deontic logic [15], data network monitoring [22], robotics [1], quantum mechanics [13] and quantum information theory [2],

Since their introduction numerous formalizations of paraconsistent logic have emerged [6, 16, 23]. This paper takes a specific standpoint: the notion of paraconsistent transition systems (PLTS) introduced in [18] and later used to reason about decoherence in quantum circuits as documented in [5]. Informally, in these systems, each transition comes with a pair of weights: one weights the evidence that the transition may occur, the other the evidence it will not.

This paper aims at developing an algebraic counterpart to our research on PLTS, already documented in a number of references [5, 17, 18, 20, 21]. Therefore, we propose an extension of KAT to a paraconsistent framework able to reason about uncertain or inconsistent computations. This allows computations to yield outcomes graded by two weights: one indicating evidence for execution and the other evidence for failure. Note that a similar motivation can be found in [25] where a graded variant of KAT is introduced.

That work, however, only captures forms of uncertainity as usual in fuzzy logic.

Paper structure. Subsection 1.1 recalls the definition of KA and KAT. Section 2 revisits the definition of PLTS parametric on a class of residuated lattices [18] and establishes some new properties that will prove useful in the sequel. Section 3 introduces a variant of KAT for a paraconsistent context (PKAT) where programs and tests accommodate inconsistencies and vagueness. Additionally, in Section 3 we present the details of two new algebraic structures that form a PKAT: paraconsistent sets $\mathbf{Set}_{\mathbf{P}}(\mathcal{T})$ and paraconsistent relations $\mathbf{Rel}_{\mathbf{P}}(\mathcal{K},\mathcal{T})$, which are parametric over fixed twisted structures $\mathcal{K}$ and $\mathcal{T}$, respectively. Finally, Section 4 concludes and points out a number of topics for future research.

1.1 Preliminaries

Definition 1. *[30] A Kleene algebra (KA) is an algebraic structure* $(K,+,\cdot,^\star,0,1)$ *satisfying the axioms* (1)-(13) *below. The order of precedence of the operators is* $^\star > \cdot > +$*. Thus,* $p+q\cdot r^\star$ *should be parsed as* $p+(q\cdot(r^\star))$.

$$p+(q+r) = (p+q)+r \quad (1)$$
$$p+q = q+p \quad (2)$$
$$p+0 = p \quad (3)$$
$$p+p = p \quad (4)$$
$$p\cdot(q\cdot r) = (p\cdot q)\cdot r \quad (5)$$
$$1\cdot p = p\cdot 1 = p \quad (6)$$
$$p\cdot(q+r) = p\cdot q+p\cdot r \quad (7)$$
$$(p+q)\cdot r = p\cdot r+q\cdot r \quad (8)$$
$$0\cdot p = p\cdot 0 = 0 \quad (9)$$
$$1+p\cdot p^\star = p^\star \quad (10)$$
$$1+p^\star\cdot p = p^\star \quad (11)$$
$$p\cdot r \leq r \longrightarrow p^\star\cdot r \leq r \quad (12)$$
$$r\cdot p \leq r \longrightarrow r\cdot p^\star \leq r \quad (13)$$

where $\leq$ *refers to the natural partial order on K, that is,* $p\leq q$ *if and only if* $p+q=q$.

Axioms (1)-(9) establish $(K,+,\cdot,^\star,0,1)$ as an idempotent semiring, while axioms (10)-(13) say that * is like the reflexive transitive closure on binary relations [26].

Definition 2. *[26] A* Kleene algebra with tests *(KAT) is a two-sorted algebra*

$$(K,B,+,\cdot,^\star,^-,0,1)$$

such that $(K,+,\cdot,^\star,0,1)$ *is a Kleene algebra,* $(B,+,\cdot,^-,0,1)$ *is a Boolean algebra and* $B\subseteq K$.

The unary operator $^-$ is defined only on B which elements are called tests. We reserve the letters p, q, r, s for arbitrary elements of K and α, β, γ for tests. In summary, a KAT satisfies axioms (1)-(13) and the following for any tests:

$$\alpha+(\beta\cdot\gamma) = (\alpha+\beta)\cdot(\alpha+\gamma) \quad (14)$$
$$(\alpha\cdot\beta)+\gamma = (\alpha+\gamma)\cdot(\beta+\gamma) \quad (15)$$
$$\alpha\cdot\beta = \beta\cdot\alpha \quad (16)$$
$$\alpha\cdot\alpha = \alpha \quad (17)$$
$$\overline{\overline{\alpha}} = \alpha \quad (18)$$
$$\alpha+1 = 1 \quad (19)$$
$$\alpha\cdot\overline{\alpha} = 0 \quad (20)$$
$$\alpha+\overline{\alpha} = 1 \quad (21)$$

Axioms (1)-(13) pertain to the fact that $(K,+,\cdot,^\star,0,1)$ is a KA. While axioms (14)-(21) pertain to the fact that $(B,+,\cdot,^-,0,1)$ is a Boolean algebra [26].

Example 1. *Semantically, programs are represented as binary relations over a set of states X and a test* α *is interpreted as a subset of the identity relation, comprising all pairs* (x,x) *such that* α *holds at state x. Hence,the family of binary relations on a set X is a KAT with operations defined as follows.*

$$0 \coloneqq \emptyset \qquad R + R' \coloneqq R \cup R'$$
$$1 \coloneqq \{(u,u) \mid u \in X\} \qquad R \cdot R' \coloneqq \{(u,w) \mid \exists v (u,v) \in R \wedge (v,w) \in R'\}$$
$$\overline{\alpha} = 1 \setminus \alpha \qquad R^\star \coloneqq \bigcup_{n \geq 0} R^n = \textit{reflexive transitive closure of } R$$

where $R^0 \coloneqq \{(u,u) \mid u \in X\}$ *and* $R^{n+1} = R \cdot R^n$.

This extension of KA with a Boolean algebra results in an algebraic model to capture program behavior and assertions. Hence, conditionals and **while** loops found in programming can be defined in terms of the regular operators.

$$\textbf{if } \alpha \textbf{ then } p \textbf{ else } q \stackrel{\text{def}}{=} \alpha \cdot p + \overline{\alpha} \cdot q$$
$$\textbf{while } \alpha \textbf{ do } p \stackrel{\text{def}}{=} (\alpha \cdot p)^\star \cdot \overline{\alpha}$$

2 Paraconsistent transition systems

The notion of *paraconsistent transition systems*, abbreviated to PLTS, was introduced in [18]. These systems' transitions involve two weights: a *positive* and a *negative* that characterize each transition in opposite ways. The positive weight represents the evidence of its presence, while the negative weight represents the evidence of its absence. Furthermore, following the line of research outlined in [9], a residuated lattice over a set A of possible truth values is adopted. This allows the transitions of PLTS to be represented by pairs of weights $(tt, ff) \in A \times A$. Thus, all the relevant constructions of PLTS are parametric in a class of residuated lattices and admit different instances according to the truth values domain A that better suits each concrete problem.

To exemplify suppose, for example, that weights for both transitions come from a residuated lattice over the real interval $[0,1]$.

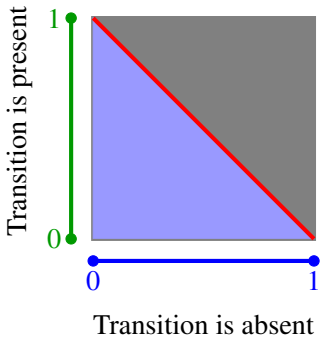


Figure 1: The vagueness-inconsistency square

These pairs of weights express different behaviours:

- *inconsistency*, when the positive and negative weights are contradictory, i.e. they sum to a value greater than 1, this corresponds to the upper triangle in Figure 1, filled in grey.
- *vagueness*, when the sum is less than 1, corresponding to the lower, periwinkle triangle in Figure 1.

- *consistency*, when the sum is exactly 1, that is the evidence degrees that enforce and prevent a transition from occuring are complementary, corresponding to the red line in Figure 1.

Our focus is on a particular class of residuated lattices that are bounded by a maximal 1 and a minimal element 0 and where the lattice meet ($\sqcap$) and the monoidal composition ($\odot$) coincide. As these two operators coincide, we will henceforth omit one when working with the residuated lattices. Thus, adjunction of a residuated lattice $\boldsymbol{A} = \langle A, \sqcap, \sqcup, 1, 0, \rightarrow \rangle$ over a set A is stated as $a \sqcap b \leq c$ if and only if $b \leq a \rightarrow c$. Additionally, a pre-linearity condition is imposed, $(a \rightarrow b) \sqcup (b \rightarrow a) = 1$

A residuated lattice satisfying pre-linearity is referred to as a *MTL-algebra* [24]. We will use the term *iMTL-algebra* for the class of *integral* semantic structures considered in the sequel, that is, pre-linear, residuated lattices such that $\sqcap$ and $\odot$ coincide.

Example 2. *The following lattices are iMTL-algebras:*

- *the Boolean algebra* $\mathbf{2} = \langle \{0,1\}, \wedge, \vee, 1, 0, \rightarrow \rangle$
- *the Łukasiewicz three-valued algebra* $\mathbf{3} = \langle \{\top, u, \bot\}, \wedge_3, \vee_3, \top, \bot, \rightarrow_3 \rangle$*, where*

$\wedge_3$	$\bot$	u	$\top$
$\bot$	$\bot$	$\bot$	$\bot$
u	$\bot$	u	u
$\top$	$\bot$	u	$\top$

$\vee_3$	$\bot$	u	$\top$
$\bot$	$\bot$	u	$\top$
u	u	u	$\top$
$\top$	$\top$	$\top$	$\top$

$\rightarrow_3$	$\bot$	u	$\top$
$\bot$	$\top$	$\top$	$\top$
u	u	$\top$	$\top$
$\top$	$\bot$	u	$\top$

The truth value u stands for "unknown" and assumes different notations in the literature, such as $1/2$ *or #, which are interpreted as "possibility" or "indeterminacy" [36].*

- $\boldsymbol{G} = \langle [0,1], \min, \max, 0, 1, \rightarrow \rangle$*, with* $a \rightarrow b = 1$*, if* $a \leq b$ *and* $a \rightarrow b = b$ *otherwise.*

The following lemma combines [19][Lemma 1] and delineates several essential properties of complete residuated lattices, which we will resort to prove some results in this paper. For detailed proofs of Properties (22)-(24) and Properties (25)-(26), readers are referred to [19] and [9], respectively.

Lemma 1. *Let* $\boldsymbol{A}$ *be an iMTL-algebra over a non empty set A. The following properties hold, for any* $a, a', b, b' \in A$

$$a \sqcap (b \sqcup b') = (a \sqcap b) \sqcup (a \sqcap b') \tag{22}$$

$$a \sqcup (b \sqcap b') = (a \sqcup b) \sqcap (a \sqcup b') \tag{23}$$

$$a \leq a' \text{ and } b \leq b' \text{ implies } a \sqcap b \leq a' \sqcap b' \tag{24}$$

$$a \sqcap \left(\bigsqcup_{i \in I} b_i \right) = \bigsqcup_{i \in I} (a \sqcap b_i) \tag{25}$$

$$\left(\bigsqcup_{i \in I} a_i \right) \sqcap b = \bigsqcup_{i \in I} (a_i \sqcap b) \tag{26}$$

where I is a (possibly infinite) index set.

Before providing a formal definition of PLTS, it is important to define the notion of a twisted structure. Initially introduced in Kracht's seminal work [32], a twisted structure results from the construction of the direct product of a residuated lattice $\boldsymbol{L}$ and its order-dual $\boldsymbol{L}^{\partial}$. This resulting lattice naturally possesses an involution[2] given by

$$/\!\!/(a, a') = (a', a)$$

[2]Informally speaking, the operator $/\!\!/$ will denote a involutive negation in this paper and in essence it interchanges the positive and negative weight within a pair. See Example 3 for a concrete example.

for all $a, a' \in \boldsymbol{L} \times \boldsymbol{L}^{\partial}$. Since its introduction, numerous authors have explored extensions of this structure by imposing additional properties on the residuated lattice [10, 11, 12]. Twisted structures play a fundamental role in the context of PLTS, as documented in prior work [17, 18, 20, 21], by enabling the computation of pairs of truth weights.

Definition 3. *Let $\boldsymbol{A} = \langle A, \sqcap, \sqcup, 1, 0, \rightarrow \rangle$ be an iMTL-algebra. Its corresponding $\boldsymbol{A}$-twisted structure is denoted by $\mathcal{A} = \langle A \times A, \between, \barwedge, /\!/, (0,1), (1,0) \rangle$ and defined for any pair in $A \times A$ as:*

$$\begin{aligned} /\!/(a,b) &= (b,a) \\ (a,a') \barwedge (b,b') &= (a \sqcap b, a' \sqcup b') \\ (a,a') \between (b,b') &= (a \sqcup b, a' \sqcap b') \end{aligned}$$

The order in $\boldsymbol{A}$ is lifted to $\mathcal{A}$ as $(a,a') \preccurlyeq (b,b')$ iff $a \leq b$ and $a' \geq b'$.

As before, parentheses will be frequently omitted, reserving their use for enhancing the readability and clarity of certain expressions.

In the sequel, we will refer to the following Lemma, which presents properties of the operators defined above. Most properties are evident since the product of two complete, universally distributive lattices is itself a complete, universally distributive lattice. Furthermore, the Lemma essentially states that the twisted structure is a quantale, which aligns with the inspiration from Chu's work in category theory and its specialization to quantales [40]. Nevertheless, we explicitly state the properties of interest so that we can refer to them later.

Lemma 2. *Let $\mathcal{A}$ be a twisted structure, as defined in Definition 3. The following properties hold for any $(a,a'), (b,b'), (c,c') \in A \times A$.*

$$/\!/ /\!/ (a,a') = (a,a') \tag{27}$$
$$(a,a') \between (1,0) = (1,0) \tag{28}$$
$$(a,a') \between (0,1) = (a,a') \tag{29}$$
$$(a,a') \between (a,a') = (a,a') \tag{30}$$
$$(a,a') \barwedge (a,a') = (a,a') \tag{31}$$
$$(a,a') \between (b,b') = (b,b') \between (a,a') \tag{32}$$
$$(a,a') \barwedge (b,b') = (b,b') \barwedge (a,a') \tag{33}$$
$$/\!/((a,a') \between (b,b')) = /\!/(a,a') \barwedge /\!/(b,b') \tag{34}$$
$$/\!/((a,a') \barwedge (b,b')) = /\!/(a,a') \between /\!/(b,b') \tag{35}$$
$$(a,a') \barwedge (1,0) = (1,0) \barwedge (a,a') = (a,a') \tag{36}$$
$$(a,a') \barwedge (0,1) = (0,1) \barwedge (a,a') = (0,1) \tag{37}$$
$$(a,a') \between \Big((b,b') \between (c,c') \Big) = \Big((a,a') \between (b,b') \Big) \between (c,c') \tag{38}$$
$$(a,a') \barwedge \Big((b,b') \barwedge (c,c') \Big) = \Big((a,a') \barwedge (b,b') \Big) \barwedge (c,c') \tag{39}$$
$$(a,a') \barwedge \Big((b,b') \between (c,c') \Big) = \Big((a,a') \barwedge (b,b') \Big) \between \Big((a,a') \barwedge (c,c') \Big) \tag{40}$$
$$\Big((a,a') \between (b,b') \Big) \barwedge (c,c') = \Big((a,a') \barwedge (c,c') \Big) \between \Big((b,b') \barwedge (c,c') \Big) \tag{41}$$

$$(a,a') \boxplus \Big((b,b') \boxbar (c,c') \Big) = \Big((a,a') \boxplus (b,b') \Big) \boxbar \Big((a,a') \boxplus (c,c') \Big) \tag{42}$$

$$\Big((a,a') \boxbar (b,b') \Big) \boxplus (c,c') = \Big((a,a') \boxplus (c,c') \Big) \boxbar \Big((b,b') \boxplus (c,c') \Big) \tag{43}$$

Proof.

- Property (27) follows immediately since $/\!/$ is involutive.
- Property (28) and Property (29) are a consequence of 0 and 1 being the the least and greatest element of set A, respectively. Hence,

$$(a,a') \boxplus (1,0) = (a \sqcup 1, a' \sqcap 0) = (1,0)$$
$$(a,a') \boxplus (0,1) = (a \sqcup 0, a' \sqcap 1) = (a,a')$$

- Property (30) is a consequence of operators $\sqcap$ and $\sqcup$ being idempotent, $(a,a') \boxplus (a,a') = (a \sqcup a, a' \sqcap a') = (a,a')$. Property (31) follows similarly.
- Property (32) is a consequence of operators $\sqcap$ and $\sqcup$ being commutative. Consequently, it follows $(a,a') \boxplus (b,b') = (a \sqcup b, a' \sqcap b') = (b \sqcup a, b' \sqcap a') = (b,b') \boxplus (a,a')$. The proof for Property (33) follows similarly.
- The proof of Property (34) follows directly by the definition of the operators, $/\!/((a,a') \boxplus (b,b')) = (a' \sqcap b', a \sqcup b) = (a',a) \boxbar (b',b) = /\!/(a,a') \boxbar /\!/(b,b')$. Similarly, it is possible to prove Property (35).
- Property (36) since 0 and 1 are the least and greatest element of A, respectively, it follows $(1,0) \boxbar (a,a') = (1 \sqcap a, 0 \sqcup a') = (a,a')$ and by (33) it follows $(a,a') \boxbar (1,0) = (a,a')$. The proof of Property (37) follows similarly.
- To prove Property (38) note that,

$$(a,a') \boxplus ((b,b') \boxplus (c,c')) = (a,a') \boxplus (b \sqcup c, b' \sqcap c') = (a \sqcup (b \sqcup c), a' \sqcap (b' \sqcap c'))$$
$$((a,a') \boxplus (b,b')) \boxplus (c,c') = (a \sqcup b, a' \sqcap b') \boxplus (c,c') = ((a \sqcup b) \sqcup c, (a' \sqcap b') \sqcap c')$$

  Since $\sqcup$ and $\sqcap$ are associative it follows that $a \sqcup (b \sqcup c) = (a \sqcup b) \sqcup c$ and $a' \sqcap (b' \sqcap c') = (a' \sqcap b') \sqcap c'$. Therefore, operator $\boxplus$ is also associative. Similarly, it is possible to prove Property (39), i.e. $\boxbar$ is associative.
- Property (40) is a consequence of Property (22) and (23).

$$\begin{aligned}
(a,a') \boxbar \Big((b,b') \boxplus (c,c') \Big) &= (a,a') \boxbar (b \sqcup c, b' \sqcap c') \\
&= (a \sqcap (b \sqcup c), a' \sqcup (b' \sqcap c')) \\
&= ((a \sqcap b) \sqcup (a \sqcap c), (a' \sqcup b') \sqcap (a' \sqcup c')) \\
&= (a \sqcap b, a' \sqcup b') \boxplus (a \sqcap c, a' \sqcup c') \\
&= \Big((a,a') \boxbar (b,b') \Big) \boxplus \Big((a,a') \boxbar (c,c') \Big)
\end{aligned}$$

  Similarly, it is possible to prove Property (41).

- To prove property (42) note that,

$$(a,a') \between \Big((b,b') \invamp (c,c') \Big) = (a \sqcup (b \sqcap c), a' \sqcap (b' \sqcup c'))$$
$$\Big((a,a') \between (b,b') \Big) \invamp \Big((a,a') \between (c,c') \Big) = ((a \sqcup b) \sqcap (a \sqcup c), (a' \sqcap b') \sqcup (a' \sqcap c'))$$

  Using property (22) and (23) it follows $a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$ and $a' \sqcap (b' \sqcup c') = (a' \sqcap b') \sqcup (a' \sqcap c')$. Similarly it is possible to prove Property (43).

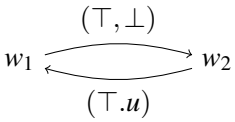
□

Finally, we present the definition of *paraconsistent transition systems* parametric to a class of residuated lattices over a set A of truth values.

Definition 4. *A paraconsistent transition system, abbreviated to PLTS, is a tuple $M = (W,R,V)$ such that:*

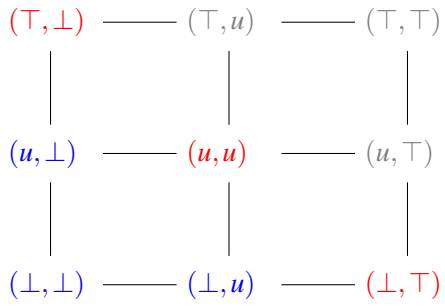
- *W is a non empty set of states,*
- *$R\colon W \times W \to A \times A$ is a paraconsistent accessibility relation. For any pair of states $(w_1,w_2) \in W \times W$ relation R assigns a pair $(tt,ff) \in A \times A$ where tt represents the evidence of the transition from w_1 to w_2 occurring and ff represents the evidence of being prevented from occurring.*
- *$V : W \times Prop \to A \times A$ is a valuation function, that assigns to a proposition symbol p at a given state w a pair $(tt,ff) \in A \times A$ such that tt is the evidence of p holding at w and ff the evidence of not holding*

Example 3. *Consider the residuated lattice* **3** *and the set of proposition symbols $\{p\}$. The following model $M = (\{w_1,w_2\},R,V)$ is a PLTS where*



$$V : W \times \{p\} \to \{\top,u,\bot\} \times \{\top,u,\bot\}$$
$$(w_1,p) \mapsto (\top,\bot)$$
$$(w_2,p) \mapsto (u,\bot)$$

We consider, following Łukasiewicz , that the truth value u is an intermediate value between $\top$ and $\bot$ [36]. Hence, a natural "ordering" of the three values is $\bot \geq u \geq \top$. In this example, each weight takes values in the set $\{\bot,u,\top\}$, making it possible to represent the resulting lattice of doing the direct product of the three-valued lattice and its order-dual.



Similar to the vagueness-inconsistency square in Figure 1, all pairs on the red anti-diagonal represent consistent information. The pairs above the anti-diagonal represent inconsistent information, while the pairs below it represent vague information. Hence, $(u,\bot)$ *denotes vague information,* $(\top,\bot)$ *consistent information and* $(\top,u)$ *inconsistent information.*

It is also worth noting that $V(w_2,p) = (u,\bot)$ *indicates that at state* w_2 *there is unknown evidence (*u*) that* p *holds and there is minimal evidence (*$\bot$*) that* p *does not hold. Similarly, it also indicates that there is minimal evidence and unknown evidence that* $\neg p$ *does and does not hold, respectively. Formally, we write*

$$V(w_2,\neg p) = /\!\!/V(w_2,p) = /\!\!/(u,\bot) = (\bot,u)$$

with $/\!\!/$ *representing the involutive negation within the framework of a twisted structure as outlined in Definition 3. We refer interested readers in the paraconsistent logic underlying PLTS to [21].*

3 Paraconsistent Kleene Algebra with tests

The approach presented in this work aims to reason about program executions in a paraconsistent manner, where executions and tests may involve vagueness as well as inconsistencies. Consequently, rather than yielding a bivalent outcome as in traditional KAT, the outcome is graded by a pair of weights, one weight indicates the evidence for execution and the other the evidence for failure. Such framework entails the need to weaken the Boolean subalgebra of KAT which leads to the following variant:

Definition 5. *A* paraconsistent Kleene algebra with tests *(PKAT) is a tuple*

$$(K,T,+,\cdot,^\star,^-,0,1)$$

such that $(K,+,\cdot,^\star,0,1)$ *is a Kleene algebra,* $(T,+,\cdot,^-,0,1)$ *satisfies axioms* (14)-(19) *and* $B \subseteq T$*. Relation* $\leq$ *is induced by* $+$*, that is,* $p \leq q$ *iff* $p+q=q$*.*

A key aspect of the KAT axiomatization lies in axioms (20) and (21), which informally express the principles of non-contradiction and excluded middle, respectively. These two principles play a significant role in the philosophy of paraconsistency, which rejects the principle of non-contradiction, and in intuitionism, which does not assume the principle of the excluded middle. The notion of PKAT takes in consideration these philosophies and consequently forms a weakened version of KAT by only rejecting axioms (20) and (21). Hence,

Theorem 1. *Any KAT is a PKAT.*

Proof. By definition, any KAT satisfies axioms (1)-(19). Thus, trivially any KAT is a PKAT. □

The weakening discussed in this paper generalizes Boolean algebras, in that Boolean algebras are precisely those algebras that satisfy both the principle of non-contradiction (20) and the principle of the excluded middle (21). This generalization is similar to that presented by Heyting algebras [8], where any Boolean algebra is a Heyting algebra that satisfies the principle of the excluded middle. In fact, the weakening presented in this paper is a stronger generalization than Heyting algebras, as any algebra failing to satisfy both principles inherently does not satisfy the principle of the excluded middle. A potential implication of this observation is that the discussed weakening of the Boolean algebra could potentially serve as algebraic models for propositional paraconsistent logic, much like how Heyting algebras model propositional intuitionistic logic [7] and Boolean algebras model propositional classical logic.

In the remaining of this section we introduce two examples of algebras where program executions and tests may encompass inconsistencies and vagueness. Consequently, these algebras can be formalized as PKAT. To achieve this, we refer back to Definition 3, which establishes that for any iMTL-algebra $\boldsymbol{K}$ over a non-empty set K of possible truth values, its corresponding twisted structure $\mathcal{K}$ allows for the computation of pairs of truth values $K \times K$.

Remark. *Given sets T and W, we denote by $(T \times T)^W$ the set of functions $W \to (T \times T)$.*

Definition 6. *Let W be a set and $\boldsymbol{T}$ be an iMTL-algebra over a non empty set of truth values T. The algebra of paraconsistent sets over the twisted-structure $\mathcal{T}$ is*

$$\mathbf{Set_P}(\mathcal{T}) = \langle (T \times T)^W, (T \times T)^W, +, \cdot, ^\star, ^-, \oslash, \Upsilon \rangle$$

For any paraconsistent sets over W, $\varphi, \psi \in (T \times T)^W$ and $w \in W$, operators are defined pointwise by

$$\begin{aligned} \Upsilon(w) &= (1,0) & (\varphi + \psi)(w) &= \varphi(w) \sqcup \psi(w) \\ \oslash(w) &= (0,1) & (\varphi \cdot \psi)(w) &= \varphi(w) \sqcap \psi(w) \\ \overline{\varphi}(w) &= /\!/\varphi(w) & (\varphi^\star)(w) &= \bigsqcup_{n \geq 0} \varphi^n(w) \end{aligned}$$

with $\varphi^0(w) = \Upsilon(w)$ and $\varphi^{n+1}(w) = (\varphi \cdot \varphi^n)(w)$. The values of paraconsistent sets $\varphi(w)$ and $\psi(w)$ are elements of $T \times T$, and constants $\oslash$ and Υ are the least and the greatest elements of $T \times T$, respectively. The partial order in paraconsistent sets φ and ψ in $(T \times T)^W$ is given by

$$\varphi \subseteq \psi \text{ if and only if } \forall w \in W,\ \varphi(w) \preccurlyeq \psi(w)$$

Example 4. *Consider the set $W = \{w_1, w_2\}$ and the three-valued residuated lattice $\mathbf{3} = \{\top, u, \bot\}$. Let $\varphi, \psi \in \mathbf{Set_P}(\mathbf{3})$ be two paraconsistent sets defined as*

$$\begin{aligned} \varphi\colon\ & W \to \mathbf{3} \times \mathbf{3} & \psi\colon\ & W \to \mathbf{3} \times \mathbf{3} \\ & w_1 \mapsto (\top, u) & & w_1 \mapsto (\top, \bot) \\ & w_2 \mapsto (u, u) & & w_2 \mapsto (\top, u) \end{aligned}$$

It is possible to define with operator $^-$ two other paraconsistent sets denoted by $\overline{\varphi}, \overline{\psi} \in \mathbf{Set_P}(\mathbf{3})$ defined as,

$$\begin{aligned} \overline{\varphi}\colon\ & W \to \mathbf{3} \times \mathbf{3} & \overline{\psi}\colon\ & W \to \mathbf{3} \times \mathbf{3} \\ & w_1 \mapsto (u, \top) & & w_1 \mapsto (\bot, \top) \\ & w_2 \mapsto (u, u) & & w_2 \mapsto (u, \top) \end{aligned}$$

Note that $\varphi \subseteq \psi$, while $\overline{\psi} \subseteq \overline{\varphi}$.

Theorem 2. *For any iMTL-algebra $\boldsymbol{T}$ over a non empty set T of possible truth values, $\mathbf{Set_P}(\mathcal{T})$ forms a PKAT.*

Proof. For a fixed iMTL-algebra $\boldsymbol{T}$, by Definition 3, we define its twisted structure $\mathcal{T}$. We will prove that $\mathbf{Set_P}(\mathcal{T})$ defined as in Definition 6 forms a PKAT, that is, we show that axioms (1)-(19) are satisfied.
Axiom (1) by Property (38), $\varphi(w) \sqcup (\psi(w) \sqcup \phi(w)) = (\varphi(w) \sqcup \psi(w)) \sqcup \phi(w)$. Using the definition of $+$, it follows that $(\varphi + (\psi + \phi))(w) = ((\varphi + \psi) + \phi)(w)$.
Axiom (2) by Property (32) and the definition of $+$, $(\varphi + \psi)(w) = (\psi + \varphi)(w)$.
Axiom (3) from Property (29) it follows that, $(\varphi + \oslash)(w) = \varphi(w) \sqcup \oslash(w) = \varphi(w) \sqcup (0,1) = \varphi(w)$.
Axiom (4) is immediate by Property (30), $(\varphi + \varphi)(w) = \varphi(w)$

Axiom (5) is a consequence of Property (39), $\varphi(w) \sqcap (\psi(w) \sqcap \phi(w)) = (\varphi(w) \sqcap \psi(w)) \sqcap \phi(w)$. Using the definition of $\cdot$, $(\varphi \cdot (\psi \cdot \phi))(w) = ((\varphi \cdot \psi) \cdot \phi)(w)$.
Axiom (6) from Property (36) it follows that $(\Upsilon \cdot \varphi)(w) = (\varphi \cdot \Upsilon)(w) = \varphi(w)$.
Axiom (7) using Property (40) and the definition of $+$ and $\cdot$ it follows that, $(\varphi \cdot (\psi + \phi))(w) = ((\varphi \cdot \psi) + (\varphi \cdot \phi))(w)$. Similarly, it is possible to show Axiom (8) that derives from Property (41).
Axiom (9) by Property (37) it follows that $(\oslash \cdot \varphi)(w) = (\varphi \cdot \oslash)(w) = \oslash(w)$.
Axiom (10) can be derived as follows,

$$\begin{aligned}(\Upsilon + (\varphi \cdot \varphi^\star))(w) =& \Upsilon(w) \sqcup \left(\varphi(w) \sqcap \bigsqcup_{n \geq 0} \varphi^n \right) \\ =& \varphi^0(w) \sqcup \left(\varphi(w) \sqcap (\varphi^0(w) \sqcup \varphi(w) \sqcup \varphi^2(w) \ldots) \right) \\ & \{\text{using (40) and defn. of } ^\star\} \\ =& \varphi^0(w) \sqcup (\varphi(w) \sqcup \varphi^2(w) \sqcup \ldots) \\ =& \varphi^\star(w)\end{aligned}$$

Similarly, by Property (41) it is possible to show Axiom (11).
Axiom (12) Let us start by assuming that $(\varphi \cdot \psi)(w) \preccurlyeq \psi(w)$. Then,

$$\begin{aligned}(\varphi^\star \cdot \psi)(w) =& \left(\varphi^0(w) \sqcup \varphi(w) \sqcup \varphi^2(w) \sqcup \ldots \right) \sqcap \psi(w) \\ & \{\text{using (36) and (41)}\} \\ =& \psi(w) \sqcup (\varphi(w) \sqcap \psi(w)) \sqcup (\varphi^2(w) \sqcap \psi(w)) \sqcup \ldots \\ =& \psi(w) \sqcup (\varphi \cdot \psi)(w) \sqcup (\varphi^2 \cdot \psi)(w) \sqcup \ldots \end{aligned} \tag{44}$$

For any integer $n \geq 0$, $(\varphi^n \cdot \psi)(w) = \underbrace{\varphi \cdot \ldots \cdot \varphi}_{n} \cdot \psi(w)$. Using the hypothesis n times, it follows that $(\varphi^n \cdot \psi)(w) \preccurlyeq \psi(w)$. Hence, by (44) and given that $(\psi \cdot \psi)(w) \preccurlyeq \psi(w)$ it follows that

$$\psi(w) \sqcup (\varphi \cdot \psi)(w) \sqcup (\varphi^2 \cdot \psi)(w) \sqcup \ldots \preccurlyeq \psi(w)$$

Showing Axiom (13) is similar but instead of using Property (41) uses (40).
Axiom (14) using Property (42) and the definition of $\cdot$ and $+$ it follows that, $(\varphi + (\psi \cdot \phi))(w) = ((\varphi + \psi) \cdot (\varphi + \phi))(w)$. Similarly, Axiom (15) follows from Property (43). Axiom (16) results from Property (33), $(\varphi \cdot \psi)(w) = (\psi \cdot \varphi)(w)$. Finally, it is possible to show Axiom (17), that is, $(\varphi \cdot \varphi)(w) = \varphi(w)$; Axiom (18), that is, $\overline{\overline{\varphi}}(w) = \varphi(w)$ and Axiom (19), that is, $(\varphi + \Upsilon)(w) = \Upsilon(w)$ follow directly from Property (31), (27) and (28), respectively. □

The aim of the following definition is to explore paraconsistent programs similar to the well-known binary programs presented in Example 1. These paraconsistent programs involve computations that may exhibit vagueness or inconsistency. Thus potentially lending themselves to representation as PLTS with transitions and valuations weighted by pairs of weights, tailored to fit the specific problem domain. Paraconsistent relations are defined over a pair of states $W \times W$, where a test α can be interpreted at any state $w \in W$. Specifically, if a test α is evaluated by a pair (tt, ff) at state w , then tt the evidence of the test holding and ff measures the evidence of the test not holding at state w. Let's proceed to define the algebra of paraconsistent relations.

Definition 7. *Let W be a set, $\boldsymbol{K}$ and $\boldsymbol{T}$ be iMTL-algebras over a non empty set of truth values K and T, respectively. The algebra of paraconsistent relations over $\mathcal{K}$ and $\mathcal{T}$ is defined as*

$$\mathbf{Rel_P}(\mathcal{K},\mathcal{T}) = \langle (K\times K)^{W\times W}, (T\times T)^{W\times W}, +, \cdot, ^\star, ^-, \oslash, \Lambda\rangle$$

where $(K\times K)^{W\times W}$ is the set of all paraconsistent relations over $W\times W$, i.e. functions $(W\times W)\to(K\times K)$. The elements of $(T\times T)^{W\times W}$ are paraconsistent tests *t such that $t(u,v) = (0,1)$ whenever $u\neq v$. The operators of paraconsistent relations are defined pointwise by*

$$\oslash(u,v) = (0,1) \qquad (R+R')(u,v) = R(u,v)\sqcup R'(u,v)$$

$$\Lambda(u,v) = \begin{cases}(1,0) & \text{if } u=v\\ (0,1) & \text{otherwise}\end{cases} \qquad (R\cdot R')(u,v) = \bigsqcup_{w\in W}(R(u,w)\sqcap R'(w,v))$$

$$\bar{t}(u,v) = /\!\!/t(u,v) \qquad R^\star(u,v) = \bigsqcup_{n\geq 0} R^n(u,v)$$

with $R^{n+1}(u,v) = (R\cdot R^n)(u,v)$ and $R^0(u,v) = \Lambda(u,v)$. The value of paraconsistent relations, $R(u,v)$ and $R'(u,v)$ are elements of $K\times K$, the value of $t(u,v)$ is an element of $T\times T$, and constants $\oslash, \Lambda$ are the least and the greatest elements of $T\times T$. The partial order $\subseteq$ for paraconsistent relations is given by

$$R\subseteq R' \text{ if and only if } \forall u,v\in W, R(u,v)\preccurlyeq R'(u,v)$$

Theorem 3. *Let $\boldsymbol{K}$ and $\boldsymbol{T}$ be iMTL-algebras over a non empty set K and T of possible truth values such that $T\subseteq K$, $\mathbf{Rel_P}(\mathcal{K},\mathcal{T})$ forms a* PKAT.

Proof. Let $\boldsymbol{K}$ and $\boldsymbol{T}$ be iMTL-algebras over set K and T, respectively, such that $T\subseteq K$. It is possible to define the corresponding twisted structures $\mathcal{K}$ and $\mathcal{T}$ as described in Definition 3. We will prove that $\mathbf{Rel_P}(\mathcal{K},\mathcal{T})$ forms a PKAT, that is, axioms (1)-(19) are satisfied. The satisfaction of axioms (1)-(4) and (18) is similar to Theorem 2. Let us show the remaining.
Axiom (5) derives as follows

$$\begin{aligned}(R\cdot(R'\cdot R''))(w,v) &= \bigsqcup_{u\in W}\left(R(w,u)\sqcap\bigsqcup_{t\in W}\left(R'(u,t)\sqcap R''(t,v)\right)\right)\\ &= \bigsqcup_{u\in W}\bigsqcup_{t\in W}\left(R(w,u)\sqcap R'(u,t)\sqcap R''(t,v)\right)\\ &= \bigsqcup_{t\in W}\left(\bigsqcup_{u\in W}\left(R(w,u)\sqcap R'(u,t)\right)\sqcap R''(t,v)\right)\\ &= \bigsqcup_{u\in W}\left((R\cdot R')(w,t)\sqcap R''(t,v)\right)\\ &= ((R\cdot R')\cdot R'')(w,v)\end{aligned}$$

Axiom (6) by definition of $\cdot$, $(\Lambda\cdot R)(w,v) = \bigsqcup_{u\in W}\left(\Lambda(w,u)\sqcap R(u,v)\right)$.
For all $u\neq w$, $\Lambda(w,u) = (0,1)$ and by (37) it follows, $\Lambda(w,u)\sqcap R(u,v) = (0,1)$. Futhermore, by Property (28) it follows that $(\Lambda\cdot R)(w,v) = \Lambda(w,w)\sqcap R(w,v) = (1,0)\sqcap R(w,v) = R(w,v)$.
Axiom (7) proceeds as

$$(R\cdot(R'+R''))(w,v) = \bigsqcup_{u\in W}(R(w,u)\sqcap(R'(u,v)\sqcup R''(u,v)))$$

{using Property (40)}

$$
\begin{aligned}
&= \bigsqcup_{u\in W}\left(\left(R(w,u)\boxcap R'(u,v)\right)\sqcup\left(R(w,u)\boxcap R''(u,v)\right)\right)\\
&= \bigsqcup_{u\in W}\left(R(w,u)\boxcap R'(u,v)\right)\sqcup\bigsqcup_{u\in W}\left(R(w,u)\boxcap R''(u,v)\right)\\
&= (R\cdot R')(w,v)+(R\cdot R'')(w,v)
\end{aligned}
$$

Axiom (8) follows similarly by replacing Property (40) with (41).
Axiom (9) by Property (37) it follows that,

$$
(\oslash\cdot R)(w,v)=\bigsqcup_{u\in W}\left(\oslash(w,u)\boxcap R(u,v)\right)=\bigsqcup_{u\in W}\left((0,1)\boxcap R(u,v)\right)=(0,1)=\oslash(w,v)
$$

Consequently, since $\boxcap$ is commutative, it follows $(R\cdot\oslash)(w,v)=\oslash(w,v)$.
Axiom (10) derives from,

$$
\begin{aligned}
(\Lambda+(R\cdot R^*))(w,v) =& \Lambda(w,v)\sqcup\left(\bigsqcup_{u\in W}R(w,u)\boxcap\left(\bigsqcup_{n\geq 0}R^n(u,v)\right)\right)\\
&\{\text{using Property (40)}\}\\
=& \Lambda(w,v)\sqcup\left(\bigsqcup_{n\geq 0}\left(\bigsqcup_{u\in W}R(w,u)\boxcap R^n(u,v)\right)\right)\\
=& R^0(w,v)\sqcup\left(\bigsqcup_{n\geq 0}(R\cdot R^n)(w,v)\right)\\
=& R^0(w,v)\sqcup\left(\bigsqcup_{n>0}R^n(w,v)\right)\\
=& R^*(w,v)
\end{aligned}
$$

Similarly, Axiom (11) follows from Property (41) and (33).
Axiom (12) Let us assume that $(R\cdot R')(w,v)\preccurlyeq R'(w,v)$. Then,

$$
\begin{aligned}
(R^*\cdot R')(w,v) =& \bigsqcup_{u\in W}\left(\bigsqcup_{n\geq 0}R^n(w,u)\boxcap R'(u,v)\right)\\
=& \bigsqcup_{n\geq 0}\left(\bigsqcup_{u\in W}(R^n(w,u)\boxcap R'(u,v))\right)\\
=& \bigsqcup_{n\geq 0}(R^n\cdot R')(w,v)\\
=& R'(w,v)\sqcup(R\cdot R')(w,v)\sqcup(R^2\cdot R')(w,v)\sqcup\ldots\\
&\{\text{Hypothesis}\}\\
\preccurlyeq& R'(w,v)
\end{aligned}
$$

Similarly, it is possible to show Axiom (13).
Axiom (14) derives as,

$$
\begin{aligned}
(t+(t'\cdot t''))(u,v) =& t(u,v)\sqcup\left(\bigsqcup_{w\in W}(t'(u,w)\boxcap t''(w,v))\right)\\
&\{(\text{step }\star)\}
\end{aligned}
$$

$$
\begin{aligned}
&= t(u,v) \mathbin{\sqcup\!\!\sqcup} \left(t'(u,v) \mathbin{\sqcap\!\!\sqcap} t''(u,v) \right) \\
&\quad \{\text{using Property (42)}\} \\
&= \left(t(u,v) \mathbin{\sqcup\!\!\sqcup} t'(u,v) \right) \mathbin{\sqcap\!\!\sqcap} \left(t(u,v) \mathbin{\sqcup\!\!\sqcup} t''(u,v) \right) \\
&= (t+t')(u,v) \mathbin{\sqcap\!\!\sqcap} (t+t'')(u,v)
\end{aligned}
$$

(step $\star$) for any $w \in W$, $(t'(u,w) \mathbin{\sqcap\!\!\sqcap} t''(w,v)) \neq (0,1)$, iff $(t'(u,w) \neq (0,1)$ and $t''(w,v) \neq (0,1))$. Thus, $(t'(u,w) \mathbin{\sqcap\!\!\sqcap} t''(w,v)) \neq (0,1)$ only when $w = u$ and $w = v$.
Also note that,

$$
\begin{aligned}
((t+t') \cdot (t+t''))(u,v) &= \bigsqcup_{w \in W} \left(\left(t(u,w) \mathbin{\sqcup\!\!\sqcup} t'(u,w) \right) \mathbin{\sqcap\!\!\sqcap} \left(t(w,v) \mathbin{\sqcup\!\!\sqcup} t''(w,v) \right) \right) \\
&\quad \{(\text{step } \star\star)\} \\
&= \left(t(u,v) \mathbin{\sqcup\!\!\sqcup} t'(u,v) \right) \mathbin{\sqcap\!\!\sqcap} \left(t(u,v) \mathbin{\sqcup\!\!\sqcup} t''(u,v) \right) \\
&= (t+t')(u,v) \mathbin{\sqcap\!\!\sqcap} (t+t'')(u,v)
\end{aligned}
$$

(step $\star\star$) for any $w \in W$,

$$
\left(t(u,w) \mathbin{\sqcup\!\!\sqcup} t'(u,w) \right) \mathbin{\sqcap\!\!\sqcap} \left(t(w,v) \mathbin{\sqcup\!\!\sqcup} t''(w,v) \right) \neq (0,1)
$$

if and only if $(t(u,w) \mathbin{\sqcup\!\!\sqcup} t'(u,w) \neq (0,1)$ and $t(w,v) \mathbin{\sqcup\!\!\sqcup} t''(w,v) \neq (0,1))$ Hence, $t(u,w)$, $t'(u,w)$, $t(w,v)$ and $t''(w,v)$ must all be different from $(0,1)$ which implies $u = w = v$.
Therefore, we show that $(t+(t' \cdot t''))(u,v) = ((t+t') \cdot (t+t''))(u,v)$. Similarly, we can show Axiom (15) by Property (43) instead of (42).
Axiom (16) by definition,

$$
\begin{aligned}
(t \cdot t')(u,v) &= \bigsqcup_{w \in W} \left(t(u,w) \mathbin{\sqcap\!\!\sqcap} t'(w,v) \right) \\
(t' \cdot t)(u,v) &= \bigsqcup_{w \in W} \left(t'(u,w) \mathbin{\sqcap\!\!\sqcap} t(w,v) \right)
\end{aligned}
$$

Whenever $w \neq u$ or $w \neq v$, by definition of test and by (37), it follows that $t(u,w) \mathbin{\sqcap\!\!\sqcap} t'(w,v) = t'(u,w) \mathbin{\sqcap\!\!\sqcap} t(w,v) = (0,1)$. Hence, $t(u,w) \mathbin{\sqcap\!\!\sqcap} t'(w,v) \neq (0,1)$ and $t'(u,w) \mathbin{\sqcap\!\!\sqcap} t(w,v) \neq (0,1)$ only when $w = u$ and $w = v$. Thus,

$$
(t \cdot t')(u,v), = t(u,v) \mathbin{\sqcap\!\!\sqcap} t'(u,v) = t'(u,v) \mathbin{\sqcap\!\!\sqcap} t(u,v) = (t' \cdot t)(u,v)
$$

Axiom (17) by definition of $\cdot$, $(t \cdot t)(u,v) = \bigsqcup_{w \in W} \left(t(u,w) \mathbin{\sqcap\!\!\sqcap} t(w,v) \right)$.
Since $t(u,w) \mathbin{\sqcap\!\!\sqcap} t(w,v) = (0,1)$ whenever $u \neq w$ or $w \neq v$. Hence, by (29) and (31) $(t \cdot t)(u,v) = t(u,v) \mathbin{\sqcap\!\!\sqcap} t(u,v) = t(u,v)$.
Axiom (19) By definition of $+$, $(t+\Lambda)(u,v) = t(u,v) + \Lambda(u,v)$. If $u = v$ then, $\Lambda(u,v) = (1,0)$ and by Property (28) $(t+\Lambda)(u,v) = (1,0) = \Lambda(u,v)$. Otherwise, $t(u,v) = (0,1)$ and by Property (29), $(t+\Lambda)(u,v) = \Lambda(u,v)$. Hence, $(t+\Lambda)(u,v) = \Lambda(u,v)$. □

4 Conclusion and future work

This paper contributes to an ongoing research agenda focused on paraconsistent transition systems (PLTS) and their logics. PLTS were initially introduced in [18], followed by the introduction of a logic to express their properties in [17] and the respective application to the analysis of quantum circuits was further discussed in [5]. Subsequently, the algebra of constructors and abstractors for PLTS was discussed in [20], which served as the foundation for a structured specification theory outlined in [21].

Here our focus is on developing an algebraic counterpart to this line of work. Hence, we take the initial steps towards introducing a variant of KAT to reason about vague and inconsistent computations and assertions. A similar roadmap for reasoning about fuzzy computations can be found in [25]. As in [25], given that such assertions often take the form of tests, our approach lies in the modification of KAT that deals with properties of tests. The approach taken in this paper rejects the principle of non-contradiction and the principle of the excluded middle; consequently, some classical properties of Boolean algebra are lost. The resulting PKAT can interpret computations entailing contradictions or vagueness.

A possible application of this work is in quantum circuits where a phenomenon known as *decoherence* can occur. Such phenomenon is characterized by the loss of information from the circuit due to unwanted interaction with the environment. When the coherence time of a qubit is exceeded, there is an increasing probability that the circuit does not behave according to its design. Typically, qubit coherence is not specified exactly but is given as time intervals in the literature, corresponding to worst-case and best-case scenarios. In [18], it is proposed to use the two accessibility relations in PLTS to model these scenarios simultaneously. The minimum coherence time $t_{\min}$ determines the negative weight, interpreted as the likelihood that the system evolves to a decoherent state, and the maximum coherence time $t_{\max}$ determines the positive weight, interpreted as the likelihood that the system remains coherent. With further considerations, it becomes feasible to translate quantum circuits into PLTS, as elaborated in [5, 18].

The parametric approach adopted in this work is based on prior work documented in [9], which has been applied to formalize paraconsistent transition systems and their corresponding logics in [17, 20]. Given two residuated lattices $\boldsymbol{K}$ and $\boldsymbol{T}$ over a set of possible truth values K and T such that $T \subseteq K$, it is possible to define a twisted structure to operate on pairs $K \times K$. This structure allows for the introduction of two algebras in this paper: $\mathbf{Set_P}(\mathcal{T})$ and $\mathbf{Rel_P}(\mathcal{K},\mathcal{T})$ parametric to the twisted structures. The main results demonstrate that both algebras $\mathbf{Set_P}(\mathcal{T})$ and $\mathbf{Rel_P}(\mathcal{K},\mathcal{T})$ form a PKAT, Theorem 2 and 3.

Since KAT provides a framework for reasoning about imperative programs in a (quasi) equational way, we aim to explore a encoding of propositional Hoare logic into PKAT. However, for such task it may be necessary to refine PKAT with additional properties and both the meaning of Hoare triples and the inference rules need adjustment. Similarly to [25], we propose encoding a Hoare triple $\{b\}p\{c\}$ in PKAT as $b \cdot p \preccurlyeq b \cdot p \cdot c$, conveying that program correctness can only improve with execution.

There is still much to be explored, such as investigating results of denesting two nested **while** loops. Additionally, it is necessary to conduct a more thorough investigation into the potential applications and limitations that may arise from the flexibility of the adopted approach.

References

[1] Jair Minoro Abe, Cláudio Rodrigo Torres, Germano Lambert-Torres, João Inácio da Silva Filho & Helga Gonzaga Martins (2007): *Paraconsistent Autonomous Mobile Robot Emmy III*. In Germano Lambert-Torres, Jair Minoro Abe, João Inácio da Silva Filho & Helga Gonzaga Martins, editors: *Advances in Technological Applications of Logical and Intelligent Systems, Selected Papers from the Sixth Congress on Logic Applied to Technology, LAPTEC 2007, Unisanta, Santa Cecilia University, Santos, Brazil, November 21-23,*

2007, *Frontiers in Artificial Intelligence and Applications* 186, IOS Press, pp. 236–258, doi:10.3233/978-1-58603-936-3-236.

[2] Juan Carlos Agudelo & Walter Alexandre Carnielli (2010): *Paraconsistent Machines and their Relation to Quantum Computing*. *J. Log. Comput.* 20(2), pp. 573–595. Available at `https://doi.org/10.48550/arXiv.0802.0150`.

[3] Seiki Akama, editor (2016): *Towards Paraconsistent Engineering*. *Intelligent Systems Reference Library* 110, Springer, doi:10.1007/978-3-319-40418-9.

[4] Roland C. Backhouse, Dexter Kozen & Bernhard Möller (2002): *Applications of Kleene Algebra (Dagstuhl Seminar 01081)*. Dagstuhl Seminar Report 298, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, doi:10.4230/DagSemRep.298.

[5] Luís Soares Barbosa & Alexandre Madeira (2023): *Capturing Qubit Decoherence through Paraconsistent Transition Systems*. In Shigeru Chiba, Youyou Cong & Elisa Gonzalez Boix, editors: *Companion Proceedings of the 7th International Conference on the Art, Science, and Engineering of Programming, Programming 2023, Tokyo, Japan, March 13-17, 2023*, ACM, pp. 109–110, doi:10.1145/3594671.3594689.

[6] DIDERIK BATENS (2001): *A GENERAL CHARACTERIZATION OF ADAPTIVE LOGICS*. *Logique et Analyse* 44(173/175), pp. 45–68. Available at `http://www.jstor.org/stable/44085197`.

[7] Nick Bezhanishvili & Dick de Jongh (2005): *Intuitionistic Logic*. *ESSLLI course notes*. Available at `https://www.math.uni-hamburg.de/en/personen/khomskii/intuitionistic/PP-2006-25.text.pdf`.

[8] Francis Borceux (1994): *Locales*, p. 1–86. Encyclopedia of Mathematics and its Applications, Cambridge University Press. Available at `https://www.cambridge.org/core/books/handbook-of-categorical-algebra/A0B8285BBA900AFE85EED8C971E0DE14`.

[9] Félix Bou, Francesc Esteva, Lluís Godo & Ricardo Oscar Rodríguez (2009): *On the Minimum Many-Valued Modal Logic over a Finite Residuated Lattice*. *Journal of Logic and Computation* 21(5), pp. 739–790, doi:10.1093/logcom/exp062.

[10] Manuela Busaniche & Roberto Cignoli (2014): *The subvariety of commutative residuated lattices represented by twist-products*. *algebra universalis* 71, doi:10.1007/s00012-014-0265-4.

[11] Manuela Busaniche, Nikolaos Galatos & Miguel Andrés Marcos (2022): *Twist Structures and Nelson Conuclei*. *Stud Logica* 110(4), pp. 949–987, doi:10.1007/S11225-022-09988-Z.

[12] Jose Castiglioni, M. Menni & Marta Sagastume (2008): *On Some Categories of Involutive Centered Residuated Lattices*. *Studia Logica* 90, pp. 93–124, doi:10.1007/s11225-008-9145-2.

[13] Maria Luisa Dalla Chiara & Roberto Giuntini (2000): *Paraconsistent ideas in quantum logic*. *Synth.* 125(1-2), pp. 55–68. Available at `https://doi.org/10.1023/A:1005296018904`.

[14] J.H. Conway (1971): *Regular Algebra and Finite Machines*. Chapman and Hall mathematics series, Chapman and Hall. Available at `https://books.google.pt/books?id=xBXvAAAAMAAJ`.

[15] Newton C. A. Costa & Walter A. Carnielli (1986): *On Paraconsistent Deontic Logic*. *Philosophia* 16(3-4), pp. 293–305, doi:10.1007/bf02379748.

[16] Newton C. A. Da Costa & E. H. Alves (1977): *A Semantical Analysis of the Calculi C N*. *Notre Dame Journal Fo Formal Logic* 18(4), pp. 621–630, doi:10.1305/ndjfl/1093888132.

[17] Ana Cruz, Alexandre Madeira & Luís Soares Barbosa (2022): *A Logic for Paraconsistent Transition Systems*. In Andrzej Indrzejczak & Michal Zawidzki, editors: *10th International Conference on Non-Classical Logics. Theory and Applications*, *EPTCS* 358, pp. 270–284. Available at `https://doi.org/10.4204/EPTCS.358.20`.

[18] Ana Cruz, Alexandre Madeira & Luís Soares Barbosa (2022): *Paraconsistent Transition Systems*. In Daniele Nantes-Sobrinho & Pascal Fontaine, editors: *Proceedings 17th International Workshop on Logical and Semantic Frameworks with Applications, LSFA 2022, Belo Horizonte, Brazil (hybrid), 23-24 September 2022*, *EPTCS* 376, pp. 3–15. Available at `https://doi.org/10.4204/EPTCS.376.3`.

[19] Juliana Cunha, Alexandre Madeira & Luis S. Barbosa (Available here): *Paraconsistent transition structures: compositional principles and a modal logic*. *(submitted to a journal)*.

[20] Juliana Cunha, Alexandre Madeira & Luís Soares Barbosa (2023): *Stepwise Development of Paraconsistent Processes*. In Cristina David & Meng Sun, editors: *Theoretical Aspects of Software Engineering - 17th International Symposium, TASE 2023, Bristol, UK, July 4-6, 2023, Proceedings*, *Lecture Notes in Computer Science* 13931, Springer, pp. 327–343. Available at `https://doi.org/10.1007/978-3-031-35257-7_20`.

[21] Juliana Cunha, Alexandre Madeira & Luís Soares Barbosa (2023): *Structured Specification of Paraconsistent Transition Systems*. In Hossein Hojjat & Erika Ábrahám, editors: *Fundamentals of Software Engineering - 10th International Conference, FSEN 2023, Tehran, Iran, May 4-5, 2023, Revised Selected Papers*, *Lecture Notes in Computer Science* 14155, Springer, pp. 1–17. Available at `https://doi.org/10.1007/978-3-031-42441-0_1`.

[22] Hyghor Miranda Côrtes, Paulo Eduardo Santos & João Inácio da Silva Filho (2022): *Monitoring electrical systems data-network equipment by means of Fuzzy and Paraconsistent Annotated Logic*. *Expert Systems with Applications* 187, p. 115865. Available at `https://doi.org/10.1016/j.eswa.2021.115865`.

[23] Michael Dunn & Greg Restall (2002): *Relevance Logic*. In D. Gabbay & F. Guenthner, editors: *Handbook of Philosophical Logic*, Kluwer Academic Publishers, pp. 1–128. Available at `https://doi.org/10.1007/978-94-017-0452-6`.

[24] Francesc Esteva & Lluis Godo (2001): *Monoidal t-norm Based Logic: Towards a Logic for Left-continuous t-norms*. *Fuzzy Sets and Systems* 124, pp. 271–288, doi:10.1016/S0165-0114(01)00098-7.

[25] Leandro Gomes, Alexandre Madeira & Luís Soares Barbosa (2019): *Generalising KAT to verify weighted computations*. *CoRR* abs/1911.01146. Available at `http://arxiv.org/abs/1911.01146`.

[26] David Harel, Jerzy Tiuryn & Dexter Kozen (2000): *Dynamic Logic*. MIT Press, Cambridge, MA, USA.

[27] S. C. Kleene (1956): *Representation of Events in Nerve Nets and Finite Automata*, pp. 3–42. Princeton University Press, Princeton. Available at `https://doi.org/10.1515/9781400882618-002`.

[28] D. Kozen (1992): *The Design and Analysis of Algorithms*. Monographs in Computer Science, Springer New York, NY. Available at `https://doi.org/10.1007/978-1-4612-4400-4`.

[29] D. Kozen (1994): *A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events*. *Information and Computation* 110(2), pp. 366–390, doi:https://doi.org/10.1006/inco.1994.1037. Available at `https://www.sciencedirect.com/science/article/pii/S0890540184710376`.

[30] Dexter Kozen (1997): *Kleene Algebra with Tests*. *ACM Trans. Program. Lang. Syst.* 19(3), pp. 427–443, doi:10.1145/256167.256195.

[31] Dexter Kozen & Maria-Cristina Patron (2000): *Certification of Compiler Optimizations Using Kleene Algebra with Tests*. In John Lloyd, Veronica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Luís Moniz Pereira, Yehoshua Sagiv & Peter J. Stuckey, editors: *Computational Logic — CL 2000*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 568–582. Available at `https://doi.org/10.1007/3-540-44957-4_38`.

[32] Marcus Kracht (1998): *On Extensions of Intermediate Logics by Strong Negation*. *Journal of Philosophical Logic* 27(1), pp. 49–73, doi:10.1023/A:1004222213212.

[33] W. Kuich & A. Salomaa (1986): *Semirings, Automata, Languages*. EATCS monographs on theoretical computer science, Springer-Verlag. Available at `https://doi.org/10.1007/978-3-642-69959-7`.

[34] Vaughan Pratt (1990): *Dynamic algebras as a well-behaved fragment of relation algebras*. In Clifford H. Bergman, Roger D. Maddux & Don L. Pigozzi, editors: *Algebraic Logic and Universal Algebra in Computer Science*, Springer New York, New York, NY, pp. 77–110. Available at `https://doi.org/10.1007/BFb0043079`.

[35] John Preskill (2018): *Quantum Computing in the NISQ era and beyond*. *Quantum* 2, p. 79, doi:10.22331/q-2018-08-06-79.

[36] Graham Priest (2007): *Paraconsistency and dialetheism*. In Dov M. Gabbay & John Woods, editors: *The Many Valued and Nonmonotonic Turn in Logic*, *Handbook of the History of Logic* 8, Elsevier, pp. 129–204. Available at `https://doi.org/10.1016/S1874-5857(07)80006-9`.

[37] Valentin N Redko (1964): *On defining relations for the algebra of regular events*. *Ukrainskii Matematicheskii Zhurnal* 16(1).

[38] Arto Salomaa (1966): *Two Complete Axiom Systems for the Algebra of Regular Events*. *J. ACM* 13(1), pp. 158–169, doi:10.1145/321312.321326. Available at `https://doi.org/10.1145/321312.321326`.

[39] Alfred Tarski (1941): *On the Calculus of Relations*. *The Journal of Symbolic Logic* 6(3), pp. 73–89. Available at `http://www.jstor.org/stable/2268577`.

[40] Constantine Tsinakis & Annika M. Wille (2006): *Minimal Varieties of Involutive Residuated Lattices*. *Studia Logica* 83(1), pp. 407–423, doi:10.1007/s11225-006-8311-7. Available at `https://doi.org/10.1007/s11225-006-8311-7`.