*Mathematical Structures in Computer Science* (2024), 1–16
doi:10.1017/S0960129524000045

CAMBRIDGE UNIVERSITY PRESS

PAPER

# Implicational Kleene algebra with domain and the substructural logic of partial correctness

Igor Sedlár

The Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic
Email: sedlar@cs.cas.cz

(Received 21 March 2023; revised 8 November 2023; accepted 19 January 2024)

**Abstract**

We show that Kozen and Tiuryn's substructural logic of partial correctness S embeds into the equational theory of Kleene algebra with domain, KAD. We provide an implicational formulation of KAD which sets S in the context of implicational extensions of Kleene algebra.

**Keywords:** Implication; Kleene algebra; partial correctness; substructural logic

## 1. Introduction

Kleene algebra with tests (Kozen 1997), KAT, is an algebraic framework for reasoning about equivalence and correctness of imperative programs. KAT comprises two-sorted algebras with a Boolean algebra of tests embedded into a Kleene algebra of programs. A Hoare-style partial correctness assertion $\{b\}\, p\, \{c\}$, meaning that condition $c$ holds after each terminating execution of program $p$ starting in a state satisfying $b$, is represented in KAT by the equation $bp\bar{c} = 0$ or, equivalently, $bp = bpc$. The equational theory of KAT is PSPACE-complete (Cohen et al. 1996). The quasi-equational theory of KAT is undecidable (Kozen 2002), but its fragment consisting of quasi-equations where the assumptions are all of the form $r = 0$ embeds into the equational theory of KAT (Kozen and Smith 1997) and is PSPACE-complete as well.

Kozen and Tiuryn (2003) extend the language of KAT with an implication operator $\Rightarrow$ and show that, in the resulting system S, partial correctness assertions can be formalized as implicational formulas $bp \Rightarrow c$. They argue that the implicational rendering of partial correctness assertions has certain advantages over the equational one, for example, it facilitates a better distinction between local and global properties. Kozen and Tiuryn formulate a sequent system for S which bears some resemblance to sequent calculi for *substructural logics* (Galatos et al. 2007). They show that their implication connective $\Rightarrow$ is similar to implication in some well-known substructural logics, but that it has also many specific features not common in substructural logic. Particular combinations of Kleene algebra and substructural logics were studied in numerous works (Buszkowski 2006; Jipsen 2004; Kozen 1994*b*; Kuznetsov 2021; Palka 2007; Pratt 1991). It is therefore interesting to look at the exact nature of the relation between S and these combinations. For one, the latter are usually undecidable (Buszkowski 2006; Kuznetsov 2021; Palka 2007), whereas S is PSPACE-complete (Kozen 2003).

In the conference paper (Sedlár and Wannenburg 2022), we showed that S embeds into a specific combination of residuated Kleene algebra and Kleene algebra with domain, KAD (Desharnais et al. 2006; Desharnais and Struth 2011). In this paper, we improve on this embedding result in

two respects. First, we show that S embeds into KAD itself. Hence, it is not necessary to use residuated implication connectives and the adjoint to the (co)domain operator as we did in Sedlár and Wannenburg (2022), and it is not even necessary to assume $^*$-continuity of KAD to obtain the result. Second, we show that KAD has an equivalent implicational formulation, iKAD, where the antidomain operator of KAD is represented using an implication operator and the constant 0. This perspective on KAD is useful from the viewpoint of the original motivation of Sedlár and Wannenburg (2022) which was to describe the relation of S and implicational extensions of Kleene algebra. Moreover, owing to the fact that KAD is decidable (EXPTIME-complete, as shown in Sedlár 2023), we obtain a (perhaps rare) example of a decidable implicational extension of Kleene algebra.

The paper is organized as follows. Section 2 recalls S and discusses its relation to Groenendijk and Stokhof's Dynamic Predicate Logic (DPL) (Groenendijk and Stokhof 1991) and Bochman and Gabbay's Sequential Dynamic Logic (SDL) (Bochman and Gabbay 2012). Section 3 shows that S embeds into the equational theory of KAD. Section 4 puts forward iKAD, the implicational formulation of KAD, and discusses the main differences between iKAD and the standard implicational extensions of Kleene algebra.

## 2. KAT and S

In this section, we recall KAT (Section 2.1) and we outline Kozen and Tiuryn's logic S (Section 2.2). We observe a connection between S and Groenendijk and Stokhof's DPL in Section 2.3, and we note that S is a fragment of Bochman and Gabbay's SDL in Section 2.4.

### *2.1 Kleene algebra with tests*

This section recalls some basic information about Kleene algebra with tests (Kozen 1997; Kozen and Smith 1997). We assume that the reader is familiar with the notion of an *idempotent semiring*.

**Definition 1.** A *Kleene algebra* (Kozen 1994*a*) is an idempotent semiring $(K, +, \cdot, 0, 1)$ expanded with an operation $^* : K \to K$ such that

$$1 + xx^* \leq x^* \tag{1}$$
$$1 + x^*x \leq x^* \tag{2}$$
$$y + xz \leq z \implies x^*y \leq z \tag{3}$$
$$y + zx \leq z \implies yx^* \leq z \tag{4}$$

(We define $x \leq y$ as $x + y = y$ and we write $xy$ instead of $x \cdot y$.)

It follows from the definition that $x^*$ is the least element $z$ such that $1 \leq z$, $xz \leq z$ and $zx \leq z$. A standard example of a Kleene algebra is a relational Kleene algebra where $K$ is a set of binary relations over some set $S$, $\cdot$ is relational composition, $+$ is set union, $^*$ is reflexive transitive closure, 1 is the identity relation on $S$, and 0 is the empty set. Another standard example is the Kleene algebra of regular languages over a finite alphabet where $\cdot$ is concatenation of languages and $^*$ is finite iteration (Kleene star).

**Definition 2.** A *Kleene algebra with tests* (Kozen 1997) is a structure of the form:

$$(K, B, +, \cdot, {}^*, {}^-, 0, 1)\,,$$

where

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

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

(That is, we assume that $B$ is closed under the semiring operations; the negation operator $^-$ is a partial function defined only on $B$.)

Every Kleene algebra is a Kleene algebra with tests; take $B = \{0, 1\}$ and define $\bar{0} = 1$, $\bar{1} = 0$. A standard example of a Kleene algebra with tests is a relational Kleene algebra expanded with a Boolean subalgebra of the *negative cone*, that is, the elements $x \leq 1$, also called subidentities, which in the relational case are subsets of the identity relation. The class of Kleene algebras with tests is denoted as KAT.

Kleene algebras with tests are able to represent *while programs* and facilitate equational reasoning about their properties such as partial correctness and equivalence:

- **skip** $= 1$
- $p\,;q = pq$
- **if** $b$ **then** $p$ **else** $q = (bp) + (\bar{b}q)$
- **while** $b$ **do** $p = (bp)^*\bar{b}$
- $\{b\}\,p\,\{c\}$ corresponds to $bp\bar{c} = 0$ (equivalently, $bp = bpc$)

The reader is referred to Kozen (1997) and references therein for more details.

### 2.2 Substructural logic of partial correctness

This section outlines the logic S (Kozen and Tiuryn 2003). As with KAT, the logic S is many-sorted. Let $\mathsf{B} = \{\mathtt{b}_i \mid i \in \omega\}$ be the set of test variables and let $\mathsf{P} = \{\mathtt{p}_i \mid i \in \omega\}$ be the set of program variables. The language of S, $\mathscr{L}_\mathsf{S}$, consists of the following sorts of syntactic objects:

| | |
|---|---|
| tests | $b, c := \mathtt{b}_i \mid 0 \mid b \Rightarrow c$ |
| programs | $p, q := \mathtt{p}_i \mid b \mid p \oplus q \mid p \otimes q \mid p^+$ |
| formulas | $e, f := b \mid p \Rightarrow f$ |
| environments | $\Gamma, \Delta := \epsilon \mid \Gamma, p \mid \Gamma, f$ |
| sequents | $\Gamma \vdash f$ |

We define $1 := 0 \Rightarrow 0$, $\neg b := b \Rightarrow 0$ and $p^* := 1 \oplus p^+$. We will sometimes write $pq$ instead of $p \otimes q$ and $\bar{b}$ instead of $\neg b$. Let $\mathsf{E} = \mathsf{B} \cup \mathsf{P}$ and let $\mathscr{E}_\mathsf{S}$, the set of S-*expressions*, be the union of the sets of formulas, programs, and environments.

Kozen and Tiuryn (2003) introduce three kinds of semantics for their language: semantics based on guarded strings, traces, and binary relations, respectively. We will work only with binary relational semantics.

**Definition 3.** An S-*model* is a pair $M = (W, V)$, where $W$ is a non-empty set and $V : \mathsf{E} \to 2^{W \times W}$ such that $V(\mathtt{b}) \subseteq \mathrm{id}_W$ for all $\mathtt{b} \in \mathsf{B}$.

For each S-model $M$, we define the $M$-*interpretation* function $[\,]_M : \mathscr{E}_\mathsf{S} \to 2^{W \times W}$ as follows:

- $[\mathtt{b}]_M = V(\mathtt{b})$
- $[\mathtt{p}]_M = V(\mathtt{p})$
- $[0]_M = \emptyset$
- $[b \Rightarrow c]_M = \{(s, s) \mid (s, s) \notin [b]_M \text{ or } (s, s) \in [c]_M\}$
- $[p \oplus q]_M = [p]_M \cup [q]_M$

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

$$\text{(Id)}\ b \vdash b \qquad \text{(I0)}\ \Gamma, 0, \Delta \vdash f$$

$$\text{(TC)}\ \frac{\Gamma, b, \Delta \vdash f \qquad \Gamma, \bar{b}, \Delta \vdash f}{\Gamma, \Delta \vdash f} \qquad \text{(cut)}\ \frac{\Gamma \vdash g \qquad \Gamma, g, \Delta \vdash f}{\Gamma, \Delta \vdash f}$$

$$\text{(R}\Rightarrow\text{)}\ \frac{\Gamma, p \vdash f}{\Gamma \vdash p \Rightarrow f} \qquad \text{(I}\Rightarrow\text{)}\ \frac{\Gamma, p, f, \Delta \vdash g}{\Gamma, p \Rightarrow f, p, \Delta \vdash g}$$

$$\text{(I}\otimes\text{)}\ \frac{\Gamma, p, q, \Delta \vdash f}{\Gamma, p \otimes q, \Delta \vdash f} \qquad \text{(E}\otimes\text{)}\ \frac{\Gamma, p \otimes q, \Delta \vdash f}{\Gamma, p, q, \Delta \vdash f}$$

$$\text{(I}\oplus\text{)}\ \frac{\Gamma, p, \Delta \vdash f \qquad \Gamma, q, \Delta \vdash f}{\Gamma, p \oplus q, \Delta \vdash f} \qquad \text{(E}\oplus_1\text{)}\ \frac{\Gamma, p \oplus q, \Delta \vdash f}{\Gamma, p, \Delta \vdash f}$$

$$\text{(I}^+\text{)}\ \frac{g, p \vdash f \qquad g, p \vdash g}{g, p^+ \vdash f} \qquad \text{(E}\oplus_2\text{)}\ \frac{\Gamma, p \oplus q, \Delta \vdash f}{\Gamma, q, \Delta \vdash f}$$

$$\text{(E}^+\text{)}\ \frac{\Gamma, p^+, \Delta \vdash f}{\Gamma, p, \Delta \vdash f} \qquad \text{(W}f\text{)}\ \frac{\Gamma, \Delta \vdash g}{\Gamma, f, \Delta \vdash g}$$

$$\text{(CC}^+\text{)}\ \frac{\Gamma, p^+, \Delta \vdash f}{\Gamma, p^+, p^+, \Delta \vdash f} \qquad \text{(W}p\text{)}\ \frac{\Gamma \vdash f}{p, \Gamma \vdash f}$$

**Figure 1.** The sequent proof system for S.

- $[p \otimes q]_M = [p]_M \circ [q]_M$
- $[p^+]_M = [p]_M^+$
- $[p \Rightarrow f]_M = \{(s, s) \mid \forall t.(s, t) \in [p]_M \implies (t, t) \in [f]_M\}$
- $[\varepsilon]_M = \mathrm{id}_W$
- $[\Gamma, \Delta]_M = [\Gamma]_M \circ [\Delta]_M$

(The symbol $^+$ denotes transitive closure and $\circ$ denotes relational composition.) A sequent $\Gamma \vdash f$ is *valid in* $M$ iff, for all $s, t \in W$, if $(s, t) \in [\Gamma]_M$, then $(t, t) \in [f]_M$ (notation: $\Gamma \vdash_M f$).

Observe that $[f]_M \subseteq \mathrm{id}_W$ for all formulas $f$; if $(s, s) \in [f]_M$, then we may say that formula $f$ is true in $s$. Note that $[bp \Rightarrow c]_M$ is the set of $(s, s)$ such that, for all $t$, if $(s, s) \in [b]_M$ and $(s, t) \in [p]_M$, then $(t, t) \in [c]_M$. Hence, $bp \Rightarrow c$ represents a partial correctness assertion: the formula is true in $s$ iff $b$ is true in $s$ and $p$ connects $s$ with a state $t$ only if $c$ is true in $t$.

Fig. 1 shows the sequent proof system for S. A sequent $\Gamma \vdash f$ is *provable in* S iff there is a finite sequence of sequents that ends with $\Gamma \vdash f$ each of which is either of the form (Id) or (I0) or is derived from previous sequents using some of the inference rules.

**Theorem** (Kozen and Tiuryn 2003). *$\Gamma \vdash f$ is provable in* S *iff $\Gamma \vdash f$ is valid in all* S*-models.*

It is evident from the semantics that $p \Rightarrow f$ corresponds to the (test of the) modal formula $[p]f$ of Propositional Dynamic Logic, PDL (Fischer and Ladner 1979; Harel et al. 2000). Hence, each environment $\Gamma$ corresponds to a program of PDL and it can be shown that $\Gamma \vdash f$ is provable in S iff $[\Gamma]f$ is valid in PDL. Similarly, $ep \vdash f$ is provable in S iff $e \to [p]f$ is valid in PDL. The logic S is clearly a syntactic expansion of KAT (modulo the choice of primitive operators). Kozen and Tiuryn (2003) show that KAT $\models p = q$ iff both $p \Rightarrow b \vdash q \Rightarrow b$ and $q \Rightarrow b \vdash p \Rightarrow b$ are provable in S where $b$ is a Boolean variable not occurring in $p$ or $q$.

S can be seen as a substructural logic (Galatos et al. 2007; Restall 2000). However, S contains some rules that are unusual from the substructural logic perspective, namely, the sort-specific weakening rules (W$f$) and (W$p$), and the implication-formula rules (TC) and (I$\Rightarrow$). It is therefore interesting to inquire how S relates to mainstream substructural logics. Results of the next two sections will shed some light on the matter.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

### *2.3 Dynamic consequence and dynamic implication*

It is clear from the previous section that S can be seen as an extension of KAT that has two specific features. First, S comes with a somewhat unusual notion of semantic entailment. In fact, the usual notion of entailment based on the subset relation would not make much sense in the present setting owing to the syntactic restriction on sequents allowing only formulas in the consequent. Second, the language of S contains an implication connective which differs semantically from the residuals of relational composition, as found in the relational semantics for Pratt's action logic, for example. This is related to the first point since, as the reader can easily verify, the implication connective gives rise to the following semantic deduction theorem:

$$\Gamma, p \vdash_M f \iff \Gamma \vdash_M p \Rightarrow f\,.$$

As a matter of fact, the entailment relation and the implication connective of S can be seen as restrictions of dynamic consequence and dynamic implication in DPL (Groenendijk and Stokhof 1991). In DPL, formulas of the standard first-order language are evaluated on *pairs* of valuations on a relational structure. Without going into details, we just mention the operations on binary relations that give rise to semantic interpretations of dynamic negation $\sim$ and dynamic implication $\rightarrow$ in models of DPL:

- $\sim R = \{(s,s) \mid \neg\exists t.(s,t) \in R\}$;
- $R \rightarrow Q = \{(s,s) \mid \forall t.((s,t) \in R \implies \exists u.(t,u) \in Q)\}$.

Moreover, a formula $\varphi$ entails $\psi$ in DPL iff, for all models and all valuations $s, t$, if $(s,t)$ is in the interpretation of $\varphi$, then there is $u$ such that $(t,u)$ is in the interpretation of $\psi$. It is easily seen that $R \rightarrow Q = \sim(R \circ \sim Q)$, and this observation will be important later on in Section 4.

It is clear that S uses the same semantic clause for $\Rightarrow$ as DPL does for $\rightarrow$ with the proviso of the syntactic restriction on implicational formulas in S: the consequent of an implicational formula is always a formula, that is, an expression whose semantic value is a subset of the identity relation. A similar remark applies to the comparison between the notion of entailment in S and the one in DPL. This semantic observation entails that S can be seen as a fragment of a combination of relational Kleene algebra with tests with a propositional fragment of DPL. Such a combination was studied by Bochman and Gabbay (2012).

### *2.4 Sequential Dynamic Logic*

Bochman and Gabbay's Sequential Dynamic Logic SDL* (Bochman and Gabbay 2012) combines features of relational Kleene algebra with tests and the propositional fragment of DPL. In particular, it adds the dynamic negation connective $\sim$ to KAT. Bochman and Gabbay provide a sound and weakly complete sequent system for dynamic entailment, and they observe that SDL* bears strong resemblance to S while lifting the syntactic restrictions of S. They leave a more thorough investigation of relations with S to another occasion. In this section, we formulate the semantics of SDL* and we make the straightforward observation that S corresponds to a syntactic fragment of SDL*.

Formulas of the language of SDL* are defined using the following grammar:

$$\varphi, \psi := \mathtt{p} \mid \mathtt{b} \mid 0 \mid \varphi \oplus \psi \mid \varphi \otimes \psi \mid \varphi^* \mid \sim\varphi$$

where $\mathtt{p} \in \mathsf{P}$ and $\mathtt{b} \in \mathsf{B}$. SDL*-sequents are expressions of the form $\Gamma \vdash \varphi$ where $\Gamma$ is a finite sequence of SDL*-formulas and $\varphi$ is a SDL*-formula. (Bochman and Gabbay use $\wedge$ instead of $\otimes$ and $\vee$ instead of $\oplus$. They do not use the constant 0; instead, their sequent system allows an empty conclusion.) Expressions of the language of SDL* are finite sequences of SDL*-formulas.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

**Definition 4.** An $\mathsf{SDL}^*$*-model* is a pair $M = (W, V)$, where $W$ is a non-empty set and $V : \mathsf{E} \to 2^{W \times W}$ such that $V(\mathsf{b}) \subseteq \mathrm{id}_W$ for all $\mathsf{b} \in \mathsf{B}$.

For each $\mathsf{SDL}^*$-model $M$, we define the *M-interpretation* function $[\,]_M : Fm_{\mathsf{SDL}^*} \to 2^{W \times W}$ as follows:

- $[\mathsf{b}]_M = V(\mathsf{b})$
- $[\mathsf{p}]_M = V(\mathsf{p})$
- $[0]_M = \emptyset$
- $[\varphi \oplus \psi]_M = [\varphi]_M \cup [\psi]_M$
- $[\varphi \otimes \psi]_M = [\varphi]_M \circ [\psi]_M$
- $[\varphi^*]_M = [\varphi]_M^*$
- $[\sim\varphi]_M = \{(s, s) \mid \neg\exists t.(s, t) \in [\varphi]_M\}$

(where $R^*$ is the reflexive transitive closure of relation $R$). We define $[\Gamma]_M := [\psi_1]_M \circ \ldots \circ [\psi_n]_M$ in case $\Gamma = (\psi_1, \ldots, \psi_n)$. If $\Gamma$ is empty, then $[\Gamma]_M = \mathrm{id}_W$. A sequent $\Gamma \vdash \varphi$ is *valid in* $M$ iff, for all $s, t \in W$, if $(s, t) \in [\Gamma]_M$, then there is $u$ such that $(t, u) \in [\varphi]_M$ (notation: $\Gamma \Vdash_M \varphi$).

Let $\delta : \mathscr{E}_{\mathsf{S}} \to \mathscr{E}_{\mathsf{SDL}^*}$ such that $\delta(\mathsf{p}) = \mathsf{p}$, $\delta(\mathsf{b}) = \mathsf{b}$, $\delta$ commutes with $\oplus, \otimes$ and the environment-forming comma operator, $\delta(p^+) = \delta(p) \otimes \delta(p)^*$, and

$$\delta(p \Rightarrow f) = \sim(\delta(p) \otimes \sim\delta(f)) .$$

**Proposition 5.** $\Gamma \vdash f$ *is provable in* $\mathsf{S}$ *iff* $\delta(\Gamma) \Vdash_M \delta(f)$ *for all* $\mathsf{SDL}^*$*-models* $M$.

*Proof.* It is sufficient to observe that every $\mathsf{S}$-model that is a counterexample to $\Gamma \vdash f$ can be turned into a $\mathsf{SDL}^*$-model that is a counterexample to $\delta(\Gamma) \Vdash \delta(f)$ and vice versa. We only need to note that if $F \subseteq \mathrm{id}_W$, then

$$P \Rightarrow F = P \to F = \sim(P \circ \sim F) .$$

□

Bochman and Gabbay provide a sound and weakly complete sequent system for dynamic consequence over $\mathsf{SDL}^*$ which bears strong resemblance to the sequent system for $\mathsf{S}$ and extends sequent systems for dynamic consequence over weaker languages studied earlier (Kanazawa 1993; van Benthem 1995, 1996; van der Does et al. 1997). We omit the details.

## 3. Embedding S into KAD

In this section, we outline $\mathsf{KAD}$ (Section 3.1) and we prove that the set of $\mathsf{S}$-provable sequents embeds into the equational theory of $\mathsf{KAD}$ (Section 3.2). We opt for a more instructive direct proof of the embedding result instead of proving the embedding via $\mathsf{PDL}$ or $\mathsf{SDL}^*$.

### *3.1 Kleene algebra with domain*

In this section, we recall KAD. Our discussion will be brief, and the reader is referred to Desharnais et al. (2006) and Desharnais and Struth (2011) for details. We note that we assume the one-sorted version of KAD where the domain operator is defined using the primitive antidomain operator (Desharnais and Struth 2011). An extension of $\mathsf{KAT}$ with a primitive domain operator (also called KAD at that time) is presented in Desharnais et al. (2006).

The language of $\mathsf{KAD}$, $\mathscr{L}_{\mathsf{KAD}}$, is one-sorted:

$$p, q := \mathsf{p}_i \mid 0 \mid 1 \mid p \cdot q \mid p + q \mid p^* \mid \mathsf{a}(p) .$$

We define $\mathsf{d}(p) := \mathsf{a}(\mathsf{a}(p))$. A *domain term* is a term of the form $\mathsf{d}(p)$.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

Kleene algebras with domain are expansions of Kleene algebras with a unary *antidomain* operator. The abstract definition below generalizes the properties of the dynamic negation operator on binary relations:

$$\sim R = \{(s, s) \mid \neg\exists t.(s, t) \in R\}\,.$$

We note that, in the literature on KAD, the link between this "relational antidomain" operator and dynamic negation does not seem to have been noted yet. Relational antidomain has a natural interpretation independent of the linguistic motivations of DPL: if $R$ is seen as the input–output relation determined by a program, then $\sim R$ is the input–output relation determined by the *test whether the program diverges*.

**Definition 6.** A *KAD* is a structure of the form

$$\mathbf{K} = (K, +, \cdot, {}^*, \mathsf{a}, 0, 1)$$

such that $(K, +, \cdot, {}^*, 0, 1)$ is a Kleene algebra, $\mathsf{a} : K \to K$ and the following are satisfied for all $x, y, z \in K$, assuming that $\mathsf{d}(x) := \mathsf{a}(\mathsf{a}(x))$:

$$\mathsf{a}(x)x = 0 \tag{5}$$

$$\mathsf{a}(xy) \leq \mathsf{a}(x\,\mathsf{d}(y)) \tag{6}$$

$$\mathsf{d}(x) + \mathsf{a}(x) = 1 \tag{7}$$

An equation $p = q$ is *valid* in $\mathbf{K}$ iff $v(p) = v(q)$ for all valuations $v$ (that is, all homomorphisms from $\mathscr{L}_{\mathsf{KAD}}$ into $\mathbf{K}$).

We will usually write $p \equiv_{\mathbf{K}} q$ to indicate that $p = q$ is valid in $\mathbf{K}$, and we will write $p \equiv_{\mathsf{KAD}} q$ to indicate that the equation $p = q$ belongs to the equational theory of KAD (i.e. it is valid in every algebra belonging to KAD); the latter will often be shortened to $p \equiv q$ if the class of algebras in question is clear from the context.

A standard example of a KAD is the Kleene algebra of binary relations over a set $S$ extended with the dynamic negation operation $\sim$. Note that the *relational domain* operation $D$ defined by:

$$D(R) := \sim\sim R = \{(s, s) \mid \exists t.(s, t) \in R\}$$

is related to the projection operation familiar from relational databases. If $R$ is seen as the input–output relation determined by a program, then $D(R)$ is the input–output relation determined by the *test whether the program halts*. The Kleene algebra of regular languages over a finite alphabet $\Sigma$ can be extended to a KAD by adding $\mathsf{a} : 2^{\Sigma^*} \to 2^{\Sigma^*}$ such that

$$\mathsf{a}(L) = \begin{cases} \{\varepsilon\} & \text{if } L = \emptyset \\ \emptyset & \text{otherwise.} \end{cases}$$

The quasivariety of Kleene algebras with domain will be denoted as KAD. Not every Kleene algebra expands to a KAD (Desharnais and Struth 2011), but the above example of a Kleene algebra of regular languages with domain shows that the equational theory of KAD is a conservative extension of the equational theory of KA. The equational theory of KAD is EXPTIME-complete (Sedlár 2023).

A *domain element* of a KAD is an $x$ such that $x = \mathsf{d}(y)$ for some $y$. In what follows, we indicate the assumption that a given $x$ is a domain element by writing $\hat{x}$. A similar notational convention is applied to domain terms.

In the rest of the paper, we will often use the equalities stated in the following lemma without explicit mention.

**Lemma 7.** *The following hold in each KAD:*

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

*(1)* $\mathsf{a}(0) = 1$ *and* $\mathsf{a}(1) = 0$*;*
*(2)* $\mathsf{d}(0) = 0$ *and* $\mathsf{d}(1) = 1$*;*
*(3)* $\mathsf{da}(x) = \mathsf{a}(x)$ *and* $\mathsf{dd}(x) = \mathsf{d}(x)$*;*
*(4)* $\mathsf{d}(x + y) = \mathsf{d}(x) + \mathsf{d}(y)$*;*
*(5)* $\mathsf{d}(x) \leq 1$ *and* $\mathsf{a}(x) \leq 1$*;*
*(6)* $\mathsf{a}(xy) = \mathsf{a}(x \cdot \mathsf{d}(y))$*;*
*(7)* $x \leq y$ *only if* $\mathsf{a}(y) \leq \mathsf{a}(x)$*;*
*(8)* $x = \mathsf{d}(x)x$*;*
*(9)* $\mathsf{d}(x) = 0$ *only if* $x = 0$*;*
*(10)* $x = x\hat{z}$ *iff* $x\mathsf{a}(\hat{z}) = 0$*;*
*(11)* $\hat{x}\hat{z} = \hat{z}\hat{x}$*;*

*Proof.* These are well-known facts about KAD; see Desharnais and Struth (2011). Some items are proven explicitly in the appendix. □

**Definition 8.** For all **K**, we define $x \trianglelefteq y$ as $x = xy$. We will write $p \trianglelefteq_{\mathsf{KAD}} q$ instead of $p \equiv_{\mathsf{KAD}} pq$.

The relation $\trianglelefteq$ is a generalization of the consequence relation of S (which corresponds to the case where $y$ is a domain element), but it does not coincide with dynamic consequence. The latter corresponds to $x \trianglelefteq \mathsf{d}(y)$. Note that $\trianglelefteq$ is a transitive relation and it coincides with $\leq$ on domain elements. The following lemma states some of its further useful properties.

**Lemma 9.** *The following hold in all* **K***:*

*(1) if* $x \leq y$ *and* $\hat{z} \leq \hat{u}$*, then* $y \trianglelefteq \hat{z}$ *only if* $x \trianglelefteq \hat{u}$*;*
*(2) if* $x \trianglelefteq z$ *and* $y \trianglelefteq z$*, then* $x + y \trianglelefteq z$*;*

*Proof.* (1) If $y = y\hat{z}$, then $\mathsf{d}(y \cdot \mathsf{a}(\hat{z})) = 0$ and so $\mathsf{d}(x \cdot \mathsf{a}(\hat{u})) = 0$ since $x \leq z$ and $\hat{z} \leq \hat{u}$. (2) If $x = xz$ and $y = yz$, then $x + y = xz + yz = (x + y)z$. □

### *3.2 Embedding* S *into Kleene algebra with domain*

In this section, we define a translation function $Tr$ from the language of S into the language of KAD and we show that it embeds the set of sequents provable in S into the equational theory of KAD.

**Definition 10.** Let $Tr : \mathscr{E}_{\mathsf{S}} \to \mathscr{L}_{\mathsf{KAD}}$ be defined as follows:

- $Tr(\mathsf{p}_n) = \mathsf{p}_{2n}$
- $Tr(\mathsf{b}_n) = \mathsf{d}(\mathsf{p}_{2n+1})$
- $Tr(p \Rightarrow f) = \mathsf{a}(Tr(p) \cdot \mathsf{a}(Tr(f)))$
- $Tr(p \oplus q) = Tr(p) + Tr(q)$
- $Tr(p \otimes q) = Tr(p) \cdot Tr(q)$
- $Tr(p^+) = Tr(p) \cdot Tr(p)^*$
- $Tr(\epsilon) = 1$
- $Tr(\Gamma, \Delta) = Tr(\Gamma) \cdot Tr(\Delta)$

It is easily verified that, for each formula $f \in \mathscr{E}_{\mathsf{S}}$, the term $Tr(f)$ is equivalent to a domain term; see Lemma 15 in the appendix. Moreover, note that $Tr(\bar{b}) = Tr(b \Rightarrow 0) = \mathsf{a}(Tr(b) \cdot \mathsf{a}(0))$ is equivalent to $\mathsf{a}(Tr(b))$.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

**Theorem.** *For all* $\Gamma, f \in \mathscr{E}_{\mathsf{S}}$,

$$\Gamma \vdash f \text{ is provable in } \mathsf{S} \iff Tr(\Gamma) \trianglelefteq Tr(f) \text{ in } \mathsf{KAD}\,.$$

*Proof.* To establish the implication from right to left, let us assume that $\Gamma \vdash f$ is not provable in $\mathsf{S}$. By Theorem 2.2, there is an $\mathsf{S}$-model $M = (W, V)$ and states $s, t \in W$ such that $(s, t) \in [\![\Gamma]\!]_M$ but $(t, t) \notin [\![f]\!]_M$. Equivalently,

$$[\![\Gamma]\!]_M \neq [\![\Gamma]\!]_M \circ [\![f]\!]_M\,.$$

Define $\mathbf{K}$ as the KAD over the set of all binary relations on $W$, and let $[\![\ ]\!]$ be the unique valuation on $\mathbf{K}$ such that, for all $n \in \omega$,

- $[\![\mathtt{p}_{2n}]\!] = [\![\mathtt{p}_n]\!]_M$, and
- $[\![\mathtt{p}_{2n+1}]\!] = [\![\mathtt{b}_n]\!]_M$.

It can be shown by induction on the complexity of expressions $\chi \in \mathscr{E}_{\mathsf{S}}$ that

$$[\![Tr(\chi)]\!] = [\![\chi]\!]_M\,.$$

The base case for $\mathtt{p}_n$ holds by definition and the base case for $\mathtt{b}_n$ is established by noting that $[\![\mathtt{b}_n]\!]_M = D([\![\mathtt{b}_n]\!]_M)$ since $[\![\mathtt{b}_n]\!]_M \subseteq \mathrm{id}_W$, and $D([\![\mathtt{b}_n]\!]_M) = [\![\mathsf{d}(\mathtt{p}_{2n+1})]\!]$. The induction step is uneventful, perhaps with the following exception:

$$\begin{aligned}
[\![p \Rightarrow f]\!]_M &= \sim([\![p]\!]_M \circ \sim [\![f]\!]_M) \\
&= \sim([\![Tr(p)]\!] \circ \sim [\![Tr(f)]\!]) \\
&= [\![\mathsf{a}(Tr(p) \cdot \mathsf{a}(Tr(f)))]\!] \\
&= [\![Tr(p \Rightarrow f)]\!]\,.
\end{aligned}$$

(For a more detailed justification, see Lemma 16 in the appendix.) It follows that

$$[\![Tr(\Gamma)]\!] \neq [\![Tr(\Gamma)]\!] \circ [\![Tr(f)]\!]\,.$$

Hence, $Tr(\Gamma)$ is not equivalent to $Tr(\Gamma) \cdot Tr(f)$ in $\mathsf{KAD}$.

The converse implication is established by induction on the length of proofs. Most of the cases of the inductive proof use only Kleene algebra; here, we prove the cases of the induction step containing implication. (The proof is carried out in more detail in the appendix; see Lemma 17.)

To establish the case corresponding to (TC), it is sufficient to show that

$$\frac{x\hat{u}y \trianglelefteq \hat{v} \qquad x\mathsf{a}(\hat{u})y \trianglelefteq \hat{v}}{xy \trianglelefteq \hat{v}}\,.$$

This is established using Lemma 9(2). The assumptions entail that $x\hat{u}y + x\mathsf{a}(\hat{u})y \trianglelefteq \hat{z}$, but

$$x\hat{u}y + x\mathsf{a}(\hat{u})y = x(\hat{u} + \mathsf{a}(\hat{u}))y = xy\,.$$

To establish the case corresponding to (R⇒), it is sufficient to show that

$$\frac{xy = xy\hat{u}}{x = x\mathsf{a}(y\mathsf{a}(\hat{u}))}\,.$$

If $xy = xy\hat{u}$, then the following holds. First,

$$\begin{aligned}
\mathsf{d}(x\mathsf{d}(y\mathsf{a}(\hat{u}))) &= \mathsf{d}(xy\mathsf{a}(\hat{u})) \\
&= \mathsf{d}(xy\hat{u}\mathsf{a}(\hat{u})) \\
&= 0\,,
\end{aligned}$$

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

and so $x\mathsf{d}(y\mathsf{a}(\hat{u})) = 0$. Second,

$$\begin{aligned} x &= x\mathsf{d}(y\mathsf{a}(\hat{u})) + x\mathsf{a}(y\mathsf{a}(\hat{u})) \\ &= 0 + x\mathsf{a}(y\mathsf{a}(\hat{u})) \\ &= x\mathsf{a}(y\mathsf{a}(\hat{u}))\,. \end{aligned}$$

To establish (I⇒), it is sufficient to show that

$$\mathsf{a}(x \cdot \mathsf{a}(\hat{z}))x \leq x\hat{z}\,.$$

(The desired result is then obtained using Lemma 9(1).) We reason as follows:

$$\begin{aligned} \mathsf{a}(x\mathsf{a}(\hat{z}))x &= \mathsf{a}(x\mathsf{a}(\hat{z}))(x\hat{z} + x\mathsf{a}(\hat{z})) \\ &= \mathsf{a}(x\mathsf{a}(\hat{z}))x\hat{z} + \mathsf{a}(x\mathsf{a}(\hat{z}))x\mathsf{a}(\hat{z}) \\ &= \mathsf{a}(x\mathsf{a}(\hat{z}))x\hat{z} + 0 \\ &= \mathsf{a}(x\mathsf{a}(\hat{z}))x\hat{z} \\ &\leq x\hat{z} \end{aligned}$$

This (together with the other cases discussed in the appendix) concludes the proof of Theorem 3.2. □

We note that essentially the same technique can be used to show that $\mathsf{SDL}^*$ embeds into $\mathsf{KAD}$ in the sense that $\Gamma \vdash \varphi$ is valid in all $\mathsf{SDL}^*$-models $M$ iff $\lambda(\Gamma) \trianglelefteq \mathsf{d}(\lambda(\varphi))$ is valid in $\mathsf{KAD}$ for a translation $\lambda$ that sends $\mathsf{p}_n$ to $\mathsf{p}_{2n}$ and $\mathsf{b}_n$ to $\mathsf{d}(\mathsf{p}_{2n+1})$.

## 4. Implicational Kleene Algebra with Domain

In this section, we return to the main question of Sedlár and Wannenburg (2022), namely the question of how $\mathsf{S}$ relates to substructural logics based on implicational expansions of Kleene algebra. We observe that $\mathsf{KAD}$ itself *can be seen as* an implicational expansion of Kleene algebra. In particular, we introduce an extension of Kleene algebra with an implication operator which we call implicational Kleene algebra with domain, $\mathsf{iKAD}$, and we establish a mutual embedding between the equational theories of $\mathsf{KAD}$ and $\mathsf{iKAD}$. This mutual embedding is inspired by the mutual intertranslatability of $\sim$ and $\rightarrow$ in $\mathsf{DPL}$. We discuss the similarities and differences between $\mathsf{iKAD}$ and other implicational extensions of Kleene algebra (such as Pratt's 1991 action logic).

### 4.1 Kleene algebra with dynamic implication

We observed in the previous section that $\mathsf{a}(x \cdot \mathsf{a}(\hat{z}))$ behaves like Kozen and Tiuryn's implication $x \Rightarrow \hat{z}$. From this point of view, $\mathsf{a}(x)$ is equivalent to "$x$ implies 0", that is, to $\mathsf{a}(x \cdot \mathsf{a}(0))$. This motivates the following question: Can we capture antidomain in terms of a primitive implication operator?

**Definition 11.** An *iKAD* is an algebra of the form:

$$\mathbf{K} = (K, +, \cdot, \rightarrow, {}^*, 0, 1)$$

where $(K, +, \cdot, {}^*, 0, 1)$ is a Kleene algebra and $\rightarrow$ is a binary operation satisfying the following axioms:

$$(x \rightarrow 0)x = 0 \tag{8}$$

$$(x \rightarrow y) + ((x \rightarrow y) \rightarrow 0) = 1 \tag{9}$$

$$(xy \rightarrow z) = (x \rightarrow (y \rightarrow z)) \tag{10}$$

$$(x \rightarrow y) = (x \rightarrow ((y \rightarrow 0) \rightarrow 0)) \tag{11}$$

We define $\sim x := x \rightarrow 0$.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

The guiding example of an iKAD is a relational Kleene algebra with the dynamic implication operation of DPL, namely,

$$R \to Q = \{(s,s) \mid \forall t : (s,t) \in R \implies \exists u (t,u) \in Q\}\,.$$

Dynamic implication expresses a test of the following liveness property: the program represented by $Q$ has a terminating execution starting in any final state of a terminating execution of the program represented by $R$ (we can always continue with $Q$ after $R$, as put by Hollenberg 1997). In the particular instance where $Q$ is a test, this means that $Q$ holds after every terminating execution of $R$ (partial correctness). Note that $R \to \emptyset$ boils down to dynamic negation of $R$.

A *domain element* (term) is an element of the form $\sim\sim x$ ($p$ instead of $x$). As before, we use the notation $\hat{x}$ ($\hat{p}$) to indicate that the element $x$ (term $p$) is a domain element (term).

Pratt's action logic (Pratt 1991) is a well-known implicational extension of Kleene algebra which was studied intensively in the recent decades (Buszkowski 2006; Jipsen 2004; Kozen 1994*b*; Kuznetsov 2021; Palka 2007). Action logic is based on *residuated Kleene algebras*, adding to Kleene algebras a pair of binary operations $\multimap$ and $\circ\!\!-$ such that

$$y \leq x \multimap z \iff xy \leq z \iff x \leq z \circ\!\!- y\,. \tag{12}$$

As axiom (10) shows, an "axiom version" of residuation holds for $\to$ in iKAD, bringing $\to$ close to the $\circ\!\!-$ operator of action logic. However, it can be shown that $\to$ *does not* residuate with $\cdot$. In particular, a counterexample to

$$xy \leq z \implies x \leq y \to z \tag{13}$$

is a three-element Kleene algebra consisting of the linearly ordered set of elements $0 < 1 < 2$ where $\cdot$ is commutative and $x \cdot 2 = 2$ in case $x \neq 0$, $2^* = 2$ and where the following table characterizes $\to$:

| $\to$ | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 2 | 0 | 1 | 1 |

It is clear that $2 \cdot 0 \leq 0$, but not $2 \leq 0 \to 0$. A similar counterexample to the converse of (13) exists. In general, it is clear that (13) should fail since $y \to z \leq 1$ for all $y, z$. We leave a more thorough comparison of iKAD and residuated Kleene algebras to another occasion.

### *4.2* KAD *and* iKAD

In this section, we show that KAD and iKAD are equivalent in the sense that the equational theory of one embeds into the equational theory of the other. A corollary of this result is that the equational theory of iKAD is decidable (EXPTIME-complete by Sedlár 2023). This is an interesting contrast to residuated Kleene algebras (Pratt's action logic).

**Definition 12.** Let $\tau : \mathscr{L}_{\mathsf{KAD}} \to \mathscr{L}_{\mathsf{iKAD}}$ such that

- $\tau(\mathsf{p}) = \mathsf{p}$ for all $\mathsf{p} \in \mathsf{P}$;
- $\tau$ commutes with the Kleene algebra operators;
- $\tau(\mathsf{a}(p)) = \tau(p) \to 0$.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

Let $\sigma : \mathscr{L}_{\mathsf{iKAD}} \to \mathscr{L}_{\mathsf{KAD}}$ such that

- $\sigma(\mathrm{p}) = \mathrm{p}$ for all $\mathrm{p} \in \mathsf{P}$;
- $\sigma$ commutes with the Kleene algebra operators;
- $\sigma(p \to q) = \mathsf{a}(\sigma(p) \cdot \mathsf{a}(\sigma(q)))$.

**Lemma 13.** *The following hold:*

*(1)* $\mathsf{KAD} \models p = q$ *implies* $\mathsf{iKAD} \models \tau(p) = \tau(q)$;
*(2)* $\mathsf{iKAD} \models p = q$ *implies* $\mathsf{KAD} \models \sigma(p) = \sigma(q)$;
*(3)* $\mathsf{KAD} \models p = \sigma(\tau(p))$;
*(4)* $\mathsf{iKAD} \models p = \tau(\sigma(p))$.

*Proof.* To prove the first item, it is sufficient to show that the translations of the antidomain axioms of KAD are valid in iKAD. Validity of the translations of (5) and (7) follows easily from (8) and (9), respectively. To deal with (6), it is sufficient to use the following equivalences which are established using Lemma 18 in the appendix:

$$\begin{aligned} \sim(x \cdot \sim\sim y) &= x \to \sim\sim\sim y \\ &= x \to \sim y \\ &= \sim(xy)\,. \end{aligned}$$

To prove the second item, it is sufficient to show that the translations of the implicational axioms of iKAD are valid in KAD. This is easy; see Lemma 14 in the appendix. To prove the third item, it is sufficient to observe that $\mathsf{a}(x \cdot \mathsf{a}(0)) = \mathsf{a}(x)$ holds in KAD. The final item is established easily using axiom (11). □

**Theorem.** *The following hold:*

*(1) For all* $p, q \in \mathscr{L}_{\mathsf{KAD}}$: $\mathsf{KAD} \models p = q$ *iff* $\mathsf{iKAD} \models \tau(p) = \tau(q)$;
*(2) For all* $p, q \in \mathscr{L}_{\mathsf{iKAD}}$: $\mathsf{iKAD} \models p = q$ *iff* $\mathsf{KAD} \models \sigma(p) = \sigma(q)$.

*Proof.* This is established easily using Lemma 13. □

Theorem 4.2 shows that S embeds into a specific implicational expansion of Kleene algebra, and that this expansion is decidable (EXPTIME-complete by Sedlár 2023).

## 5. Conclusion

In this sequel to the conference paper (Sedlár and Wannenburg 2022), we have shown that Kozen and Tiuryn's substructural logic of partial correctness S embeds into the equational theory of KAD. We introduced a formulation of KAD that replaces the antidomain operator a with an implication operator $\to$, thereby showing that S embeds into a particular implicational expansion of Kleene algebra. We discussed the main differences between the implicational formulation of KAD and the standard implicational expansions of Kleene algebra based on residuated semirings such as Pratt's action logic. We hope that these results contribute to a better understanding of the place of S in the wider context of substructural logics and implicational expansions of Kleene algebra.

We also noted a close relation between KAD and the propositional fragment of DPL. One aspect of this connection is the relation of S to SDL* of Bochman and Gabbay which we also commented on. We note that we could obtain our embedding result via an embedding of either PDL or SDL* into KAD, but we opted for a more instructive direct proof.

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

The connections observed in this paper and elsewhere motivate a more systematic study of algebras with (a generalization of) the dynamic negation operator. We leave such a study for another occasion.

**Acknowledgements.** The author is grateful to the reviewer for their comments. Work on this paper was supported by the long-term strategic development financing of the Institute of Computer Science of the Czech Academy of Sciences (RVO:67985807).

## References

Bochman, A. and Gabbay, D. M. (2012). Sequential dynamic logic. *Journal of Logic, Language and Information* **21** (3) 279–298.

Buszkowski, W. (2006). On action logic: equational theories of action algebras. *Journal of Logic and Computation* **17** (1) 199–217.

Cohen, E., Kozen, D. and Smith, F. (1996). The complexity of Kleene algebra with tests. Technical Report TR96-1598, Computer Science Department, Cornell University.

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

Desharnais, J. and Struth, G. (2011). Internal axioms for domain semirings. *Science of Computer Programming* **76** (3) 181–203. Special issue on the Mathematics of Program Construction (MPC 2008).

Fischer, M. J. and Ladner, R. E. (1979). Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences* **18** 194–211.

Galatos, N., Jipsen, P., Kowalski, T. and Ono, H. (2007). *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*, Elsevier, Amsterdam.

Groenendijk, J. and Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy* **14** (1) 39–100.

Harel, D., Kozen, D. and Tiuryn, J. (2000). *Dynamic Logic*, MIT Press, Cambridge, MA.

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

Jipsen, P. (2004). From semirings to residuated Kleene lattices. *Studia Logica* **76** (2) 291–303.

Kanazawa, M. (1993). Completeness and decidability of the mixed style of inference with composition. In: Dekker, P. and Stokhof, M. (eds.) *Proceedings of the Ninth Amsterdam Colloquium*, 377–390.

Kozen, D. (1994a). A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation* **110** (2) 366–390.

Kozen, D. (1994b). On action algebras. In: *Logic and Information Flow*, MIT Press, Cambridge, MA, 78–88.

Kozen, D. (1997). Kleene algebra with tests. *ACM Transactions on Programming Languages and Systems* **19** (3) 427–443.

Kozen, D. (2002). On the complexity of reasoning in Kleene algebra. *Information and Computation* **179** 152–162.

Kozen, D. (2003). Automata on guarded strings and applications. *Matématica Contemporânea* **24** 117–139.

Kozen, D. and Smith, F. (1997). Kleene algebra with tests: Completeness and decidability. In: van Dalen, D. and Bezem, M. (eds.) *Computer Science Logic*, Berlin, Heidelberg, Springer Berlin Heidelberg, 244–259.

Kozen, D. and Tiuryn, J. (2003). Substructural logic and partial correctness. *ACM Transactions on Computational Logic* **4** (3) 355–378.

Kuznetsov, S. (2021). Action logic is undecidable. *ACM Transactions on Computational Logic* **22** (2) 1–26.

Palka, E. (2007). An infinitary sequent system for the equational theory of *-continuous action lattices. *Fundamenta Informaticae* **78** (2) 295–309.

Pratt, V. (1991). Action logic and pure induction. In: *Logics in AI. European Workshop JELIA'90, Amsterdam, the Netherlands, September 10–14, 1990. Proceedings*, Berlin etc., Springer-Verlag, 97–120.

Restall, G. (2000). *An Introduction to Substrucutral Logics*, London, Routledge.

Sedlár, I. (2023). On the complexity of Kleene algebra with domain. In: Glück, R., Santocanale, L. and Winter, M. (eds.) *Relational and Algebraic Methods in Computer Science (RAMiCS 2023)*, Lecture Notes in Computer Science, vol. 13896, Cham, Springer, 208–223.

Sedlár, I. and Wannenburg, J. J. (2022). Embedding Kozen-Tiuryn logic into residuated one-sorted Kleene algebra with tests. In: Ciabattoni, A., Pimentel, E. and de Queiroz, R. J. G. B. (eds.) *Logic, Language, Information, and Computation (WoLLIC 2022)*, Lecture Notes in Computer Science, vol. 13468, Cham, Springer International Publishing, 221–236.

van Benthem, J. (1995). Logic and the flow of information. In: Prawitz, D., Skyrms, B. and Westerståhl, D. (eds.) *Logic, Methodology and Philosophy of Science IX*, Studies in Logic and the Foundations of Mathematics, vol. 134, Elsevier, Amsterdam, 693–724.

van Benthem, J. (1996). *Exploring Logical Dynamics*, CSLI Publications, Stanford, CA.

van der Does, J., Groeneveld, W. and Veltman, F. (1997). An update on "might". *Journal of Logic, Language and Information* **6** (4) 361–380.

## Appendix A. A technical appendix

This appendix contains the expanded proofs of some of the results stated (or needed) in the main text.

**Lemma 7.** *The following hold in each Kleene algebra with domain:*

*(1)* $\mathsf{a}(0) = 1$ *and* $\mathsf{a}(1) = 0$;
*(2)* $\mathsf{d}(0) = 0$ *and* $\mathsf{d}(1) = 1$;
*(3)* $\mathsf{da}(x) = \mathsf{a}(x)$ *and* $\mathsf{dd}(x) = \mathsf{d}(x)$;
*(4)* $\mathsf{d}(x + y) = \mathsf{d}(x) + \mathsf{d}(y)$;
*(5)* $\mathsf{d}(x) \leq 1$ *and* $\mathsf{a}(x) \leq 1$;
*(6)* $\mathsf{a}(xy) = \mathsf{a}(x \cdot \mathsf{d}(y))$;
*(7)* $x \leq y$ *only if* $\mathsf{a}(y) \leq \mathsf{a}(x)$;
*(8)* $x = \mathsf{d}(x)x$;
*(9)* $\mathsf{d}(x) = 0$ *only if* $x = 0$;
*(10)* $x = x\hat{z}$ *iff* $x\mathsf{a}(\hat{z}) = 0$;
*(11)* $\hat{x}\hat{z} = \hat{z}\hat{x}$;

*Proof.* We prove some of the items explicitly. (9) $\mathsf{d}(x) = 0$ only if $\mathsf{a}(x) = 1$. Then $0 = \mathsf{a}(x)x = 1x = x$. (10) If $x = x\hat{z}$, then $x\mathsf{a}(dxz) = x\hat{z}\mathsf{a}(\hat{z}) = 0$. Conversely, $x = x(\hat{z} + \mathsf{a}(\hat{z})) = x\hat{z} + x\mathsf{a}(\hat{z}) = x\hat{z} + 0 = x\hat{z}$. □

**Lemma 14.** *The following hold in each Kleene algebra with domain (where* $x \to y := \mathsf{a}(x \cdot \mathsf{a}(y))$*):*

*(1)* $(x \to 0)x = 0$;
*(2)* $(x \to y) + \mathsf{a}(x \to y) = 1$;
*(3)* $(xy \to z) = (x \to (y \to z))$;
*(4)* $(x \to y) = \mathsf{d}(x \to y)$

*Proof.* Item 1: $\mathsf{a}(x \cdot \mathsf{a}(0))x = \mathsf{a}(x1)x = \mathsf{a}(x)x = 0$. Item 2: if $z = \mathsf{a}(x \cdot \mathsf{a}(y))$, then $(x \to y) + \mathsf{a}(x \to y) = \mathsf{a}(z) + \mathsf{d}(z) = 1$. Item 3: $(xy \to z) = \mathsf{a}(xy \cdot \mathsf{a}(z)) = \mathsf{a}(x\mathsf{d}(y \cdot \mathsf{a}(z))) = x \to (y \to z))$. Item 4: $\mathsf{d}(x \to y) = \mathsf{da}(x \cdot \mathsf{a}(y)) = \mathsf{a}(x \cdot \mathsf{a}(y)) = x \to y$. □

**Lemma 15.** *For each formula* $f \in \mathscr{E}_{\mathsf{S}}$*, the term* $Tr(p)$ *is equivalent to a domain term. That is,*

$$Tr(f) \equiv_{\mathsf{KAD}} \mathsf{d}(q)$$

*for some* $q \in \mathscr{L}_{\mathsf{KAD}}$.

*Proof.* There are three cases to consider (it is not necessary to reason by induction on the complexity of $f$):

- $Tr(\mathsf{b}_n) = \mathsf{d}(\mathsf{p}_{2n+1})$
- $Tr(0) = 0 \equiv \mathsf{d}(0)$;
- $Tr(p \Rightarrow f) = \mathsf{a}(Tr(p) \cdot \mathsf{a}(Tr(f))) \equiv \mathsf{da}(Tr(p) \cdot \mathsf{a}(Tr(f)))$.

This concludes the proof. □

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

Recall the definitions of dynamic negation (relational antidomain) and the relational domain operator, for any $R \subseteq W \times W$:

$$\sim R := \{(s,s) \mid \neg\exists t : (s,t) \in R\}$$
$$D(R) := \sim\sim R = \{(s,s) \mid \exists t : (s,t) \in R\}\,.$$

**Lemma 16.** *The following hold in each* S*-model, assuming the above definitions of* $\sim$ *and D:*

*(1) for all* $f$, $[\![f]\!]_M = D[\![f]\!]_M$;
*(2) for all* $p$ *and all* $f$, $[\![p \Rightarrow f]\!]_M = \sim([\![p]\!]_M \circ \sim[\![f]\!]_M)$.

*Proof.* The first claim is obvious from the inspection of the semantic clauses for formulas, and the fact that $D(R) = R$ for $R \subseteq \mathrm{id}_W$. The second claim is established as follows (we omit the subscript $M$):

$$\begin{aligned}
\sim([\![p]\!] \circ \sim[\![f]\!]) &= \{(s,s) \mid \neg\exists t.(s,t) \in ([\![p]\!] \circ \sim[\![f]\!])\} \\
&= \{(s,s) \mid \neg\exists t.(s,t) \in [\![p]\!] \ \&\ (t,t) \in \sim[\![f]\!]\} \\
&= \{(s,s) \mid \forall t((s,t) \in [\![p]\!] \implies \exists u.(t,u) \in [\![f]\!])\} \\
&= \{(s,s) \mid \forall t((s,t) \in [\![p]\!] \implies (t,t) \in [\![f]\!])\} \\
&= [\![p \Rightarrow f]\!]
\end{aligned}$$

□

**Lemma 17.** *If* $\Gamma \vdash f$ *is provable in* S*, then*

$$Tr(\Gamma) \equiv_{\mathsf{KAD}} Tr(\Gamma) \cdot Tr(f)\,.$$

*Proof.* Induction on the length of derivations in the sequent system for S. Most rules are checked routinely, and the implicational rules are handled in the main text. Here, we add the proof related to the rule ($\mathrm{I}^+$), for which we used the assumption of $^*$-continuity in the conference paper (Sedlár and Wannenburg 2022). In particular, we show that the quasi-equation

$$e \leq 1 \ \&\ ep = epe \ \&\ ep = epq \implies ep^+ = ep^+q \tag{A1}$$

is valid in Kleene algebra. In order to show this, we use the fact that the following equation and two quasi-equations are valid in Kleene algebra:

$$p(qp)^* = (pq)^*p \tag{A2}$$
$$ep = epq \implies (ep)^+ = (ep)^+q \tag{A3}$$
$$e \leq 1 \ \&\ ep = epe \implies ep^+ = (ep)^+ \tag{A4}$$

For a proof of (A2), see Kozen (1994*a*), Corollary 2.5. The quasi-equation (A3) is established as follows:

$$ep = epq \tag{A5}$$
$$(ep)^*ep = (ep)^*epq \tag{A6}$$
$$(ep)^+ = (ep)^+q\,. \tag{A7}$$

To establish the quasi-equation (A4), we will show that

$$ep \leq (ep)^+ \tag{A8}$$
$$ep = epe \implies (ep)^+p \leq (ep)^+ \tag{A9}$$

https://doi.org/10.1017/S0960129524000045 Published online by Cambridge University Press

Then, using (4), one can infer that $ep^+ \leq (ep)^+$; conversely, one can show that

$$(ep)^+ = ep(ep)^* \leq epp^* = ep^+$$

using the assumption $e \leq 1$.

Validity of (A8) is straightforward (since $1 \leq q^*$). Validity of (A9) is established using (A2) and the assumption $ep = epe$ as follows:

$$\begin{aligned}(ep)^+p &= ep(ep)^*p \\ &= (ep)^*epp \\ &= (ep)^*epep \\ &= (ep)^+ep \\ &\leq (ep)^+\end{aligned}$$

The last step is valid since $qq^* \leq q^*$ is valid in Kleene algebra. This concludes the proof of (A9) and so (A1) is established. □

**Lemma 18.** *The following hold in each iKAD:*

*(1)* $\sim(xy) = x \to \sim y$
*(2)* $\sim\sim\sim x = \sim x$

*Proof.* Item 1: $\sim(xy) = xy \to 0 = x \to (y \to 0) = x \to \sim y$. Item 2 follows easily from axiom (11). □

**Cite this article:** Sedlár I (2024). Implicational Kleene algebra with domain and the substructural logic of partial correctness. *Mathematical Structures in Computer Science*. https://doi.org/10.1017/S0960129524000045