Information Processing Letters 116 (2016) 284–288

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On the expressive power of Kleene algebra with domain

Georg Struth

Department of Computer Science, University of Sheffield, UK

ARTICLE INFO

Article history:
Received 26 July 2015
Received in revised form 14 October 2015
Accepted 6 November 2015
Available online 11 November 2015
Communicated by J.L. Fiadeiro

Keywords:
Semirings
Kleene algebras
Expressive power
Hoare logic
Program correctness

ABSTRACT

It is shown that antidomain semirings are more expressive than test semirings and that Kleene algebras with domain are more expressive than Kleene algebras with tests. It is also shown that Kleene algebras with domain are expressive for propositional Hoare logic whereas Kleene algebras with tests are not.

1. Introduction

Kleene algebras with tests (KAT) [1] yield arguably the simplest and most elegant model of the control flow in simple while-programs. They provide an abstract algebraic view on the standard relational semantics of imperative programs, have been applied to various program analysis tasks and form the backbone of program construction and verification tools. In particular, the inference rules of propositional Hoare logic (PHL)—Hoare logic without an assignment rule—can be derived in this setting [2]. Kleene algebras with domain (KAD) [3,4] are a similar formalism that provides an algebraic approach to propositional dynamic logic and predicate transformer semantics. The inference rules of PHL are derivable in KAD as well and it is known that every KAD is a KAT [4].

From a complexity point of view, the equational theory of KAT is known to be PSPACE complete [5], whereas that of KAD is decidable in EXPTIME [6]. It seems also plausible that KAD is more expressive than KAT; after all, modal box and diamond operators can be defined in KAD.

This article makes the expressivity gap between KAT and KAD precise, showing that KAD is strictly more expressive than KAT with a simple, natural and interesting example. Firstly, I show that the inversion of the sequential composition rule of PHL, when expressed as a formula in the language of KAT, is derivable from the axioms of KAD. Secondly, I present a model of KAT in which this formula does not hold. In addition I show that KAT is not expressive for PHL, whereas this is trivially the case for KAD.

Inverting the inference rules of Hoare logic is relevant to verification condition generation in the context of program correctness, where intermediate assertions such as weakest liberal preconditions need to be computed. It is also related to the question of expressivity of Hoare logic in relative completeness proofs.

2. KAD and KAT

This section presents the four algebraic classes considered in this article. Elements of these algebras can be viewed as programs or computations. Binary relations as a model of state change provide formal support for this view.

Firstly, *antidomain semirings* are semirings with an additional antidomain operation that models those states from which a given computation is not enabled. Secondly, *test*

E-mail address: g.struth@sheffield.ac.uk.

http://dx.doi.org/10.1016/j.ipl.2015.11.007
0020-0190/© 2015 Elsevier B.V. All rights reserved.

semirings are semirings endowed with a notion of test (as in a conditional or loop), proposition or assertion. Finally, *Kleene algebras with domain* are antidomain semirings with an additional operation of finite iteration of computations (the Kleene star); and *Kleene algebras with tests* are test semirings extended by this operation. The subclass relationships between these algebras are listed in Lemma 3 and depicted in Fig. 1.

A *semiring* is a structure $(S,+,\cdot,0,1)$ such that $(S,+,0)$ is a commutative monoid, $(S,\cdot,1)$ is a monoid; and the two monoids interact via the distributivity laws

$$x\cdot(y+z)=x\cdot y+x\cdot z,\qquad (x+y)\cdot z=x\cdot z+y\cdot z$$

and the annihilation laws $0\cdot x=0$ and $x\cdot 0=0$.

A *dioid* is an additively idempotent semiring, that is, $x+x=x$ holds for all $x\in S$. In this case, $(S,+)$ forms a semilattice with order relation defined as

$$x\le y\Leftrightarrow x+y=y.$$

Multiplication is isotone with respect to the order, $x\le y$ implies both $z\cdot x\le z\cdot y$ and $x\cdot z\le y\cdot z$, and $0\le x$ holds for all $x\in S$.

A *Kleene algebra* is a structure $(K,+,\cdot,0,1,^*)$ such that $(K,+,\cdot,0,1)$ is a dioid and the star operation $^*:K\to K$ satisfies the axioms

$$1+x\cdot x^*=x^*,\qquad z+x\cdot y\le y\Rightarrow x^*\cdot z\le y,$$
$$1+x^*\cdot x=x^*,\qquad z+y\cdot x\le y\Rightarrow z\cdot x^*\le y.$$

One may think of $+$ as the nondeterministic choice between two computations, $\cdot$ as their sequential composition, 0 as the abortive computation, 1 as the ineffective computation, and * as the finite iteration of computations.

An *antidomain semiring* [4] is a structure $(S,+,\cdot,0,1,a)$ such that $(S,+,\cdot,0,1)$ is a semiring and the antidomain operation $a:S\to S$ satisfies the axioms

$$a(x)\cdot x=0,$$
$$a(x\cdot y)+a(x\cdot a(a(y)))=a(x\cdot a(a(y))),$$
$$a(x)+a(a(x))=1.$$

These imply that every antidomain semiring is a dioid. Intuitively, $a(x)$ models all those states from which computation x is not enabled. A *domain operation* can be defined on S as $d=a\circ a$. Of course, $d(x)$ models the set of all states from which x *is* enabled. The function d is a retraction, that is, $d\circ d=d$, and it follows that $x\in d(S)\Leftrightarrow d(x)=x$, where $d(S)$ denotes the image of the set S under d. Moreover, $a\circ a\circ a=a$, which implies that $d(S)=a(S)$. These facts can be used to show that $(a(S),+,\cdot,a,0,1)$ forms a boolean algebra in which multiplication coincides with meet, whereas direct axiomatisations of the domain operation would only yield a subalgebra $d(S)$ that is a distributive lattice [4]. The fixpoint property $d(x)=x$ can be used for declaring domain and antidomain elements in formulas. Since $\forall x.\, d(x)=x\Rightarrow\varphi(x)$ and $\forall x.\,\varphi(d(x))$ are logically equivalent, one can equally write $d(t)$ to denote an element of the domain or antidomain subalgebra. The antidomain operation a is boolean complementation on $a(S)$; its presence is therefore essential for the expressivity proof.

In addition, the following fact about antidomain semirings is needed.

Lemma 1. *(See [4].) In every antidomain semiring,*

$$x\cdot y=0\Leftrightarrow x\cdot d(y)=0.$$

A *Kleene algebra with domain* [4] is a structure given by $(K,+,\cdot,1,a,^*)$ such that $(K,+,\cdot,0,1,a)$ is an antidomain semiring and $(K,+,\cdot,0,1,^*)$ a Kleene algebra.

A *test semiring* is a structure (S,B,ι) such that S is a dioid, B a boolean algebra called *test algebra* or *algebra of tests*, and $\iota:B\to S$ an embedding:

$$\iota(0)=0,\qquad \iota(1)=1,\qquad \iota(x\sqcup y)=\iota(x)+\iota(y),$$
$$\iota(x\sqcap y)=\iota(x)\cdot\iota(y).$$

A *Kleene algebra with tests* [1] is a test semiring (K,B,ι) for which K is a Kleene algebra. In the tradition of Kleene algebras with tests, the embedding is left implicit. I write $p,q,r,\dots$ for tests, and x, y, z for arbitrary semiring elements and spell out the signature as $(K,B,+,\cdot,\bar{\ },0,1,^*)$. By the embedding, boolean join and semiring addition are identified and so are boolean meet and semiring multiplication. The least and greatest element of the boolean algebra are mapped to 0 and 1 of the semiring, respectively. The operation $\bar{\ }$ is partial; it denotes complementation on the boolean algebra.

I write AS for the class and axiom system of antidomain semirings, TS for that of test semirings, KAD for that of Kleene algebras with domain and KAT for that of Kleene algebras with tests.

The notions of domain, antidomain and test can be motivated from the model of binary relations.

Proposition 2. *(See [2,4].) Let $2^{A\times A}$ be the set of binary relations over the set A. Suppose that*

$$R\cdot S=\{(a,b)\mid \exists c.\ (a,c)\in R\wedge(c,b)\in S\},$$
$$id=\{(a,a)\mid a\in A\},$$
$$a(R)=\{(a,a)\mid \forall b.\ (a,b)\notin R\},$$
$$R^*=\bigcup_{i\in\mathbb{N}}R^i,$$

where $R^0=id$ and $R^{i+1}=R\cdot R^i$. Then

1. $(2^{A\times A},\{R\mid R\subseteq id\},\cup,\cdot,\bar{\ },\emptyset,id,^*)\in$ KAT,
2. $(2^{A\times A},\cup,\cdot,\emptyset,id,a,^*\}\in$ KAD.

The operation $\cdot$ on relations is the standard relational composition; *id* is the identity relation on A and * the reflexive transitive closure operation. Tests do not correspond to sets $B\subseteq A$, but to relations $\{(b,b)\mid b\in B\}\subseteq$ id, so-called *subidentities*. The operation a is domain complementation on relations; $a(R)$ represents those states in A that are not related by R to any other state as a subidentity. Hence a maps each subidentity to its boolean complement in the subalgebra of subidentities. In the relational semantics of programs, as usual, binary relations model state changes in a program as caused by assigning values to variables.