Modal Design Algebra

Walter Guttmann\textsuperscript{1} and Bernhard Möller\textsuperscript{2}

\textsuperscript{1} Abteilung Programmiermethodik und Compilerbau, Fakultät für Informatik, Universität Ulm, D-89069 Ulm, Germany
walter.guttmann@uni-ulm.de

\textsuperscript{2} Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany
moeller@informatik.uni-augsburg.de

Abstract. We give an algebraic model of the designs of UTP based on a variant of modal semirings, hence generalising the original relational model. This is intended to exhibit more clearly the algebraic principles behind UTP and to provide deeper insight into the general properties of designs, the program and specification operators, and refinement. Moreover, we set up a formal connection with general and total correctness of programs as discussed by a number of authors. Finally we show that the designs form a left semiring and even a Kleene and omega algebra. This is used to calculate closed expressions for the least and greatest fixed-point semantics of the demonic while loop that are simpler than the ones obtained from standard UTP theory and previous algebraic approaches.

1 Introduction

The Unifying Theories of Programming (UTP), developed in \cite{13}, model the termination behaviour of programs using two special variables \(ok\) and \(ok'\) that express whether a program has been started and has terminated, respectively. Specifications and programs are identified with predicates relating the initial values \(v\) of variables to their final values \(v'\); moreover, \(ok\) and \(ok'\) may occur freely in predicates. Using these variables, Hoare and He introduce a special class of predicates that reflect an assumption/commitment style of specification. These \emph{designs} have the form

\[ \begin{align*}
P \vdash Q & \iff_{\text{df}} ok \land P \Rightarrow ok' \land Q, \\
\end{align*} \]

with \(ok\) and \(ok'\) not occurring in \(P\) or \(Q\). The informal meaning is: if a computation allowed by the design has started in a state that satisfies the precondition \(P\) it will eventually terminate in a state that satisfies the postcondition \(Q\).

In the general case, UTP allows the precondition \(P\) to involve both initial and final values of the program variables. A subclass that is interesting for a number of reasons is that of \emph{normal} designs in which \(P\) is a \emph{condition}, i.e., is only allowed to depend on input values of variables. Originally \cite{13} these were called (H3) designs and characterised by a healthiness condition; the term “normal” is due to \cite{10}. A yet smaller subclass, the \emph{feasible} or (H4) designs models programs that cannot “recover” from nontermination.
The aims and results of the present paper are the following:

1. We model normal designs in a more general class of algebras than pure relation algebra. This is intended to exhibit more clearly the algebraic principles behind UTP and to provide deeper insight into the general properties of designs, the program and specification operators, and refinement.

2. We set up a formal connection between UTP and the theories of general (e.g., [2, 3, 9, 19, 21]) and total (e.g., [1, 5, 6, 8, 20]) correctness of programs (the latter also being known as demonic semantics).

3. We show that the designs form a left semiring and even a Kleene and omega algebra. This is used to calculate closed expressions for the least and greatest fixed-point semantics of the demonic while loop that are simpler than the ones obtained from standard UTP theory and previous algebraic approaches.

To achieve this we model normal designs as pairs \((a, t)\) where \(a\) corresponds to a state transition relation and condition \(t\) characterises the input states from which termination is guaranteed. The structure from which \(a\) and \(t\) are taken is that of an idempotent semiring which is an algebraic abstraction of the basic operations of choice and sequential composition, as detailed in the next section.

## 2 The Basis: Choice and Composition

A **semiring** is a structure \((S, +, 0, \cdot, 1)\) such that

- \((S, +, 0)\) is a commutative monoid,
- \((S, \cdot, 1)\) is a monoid,
- operation \(\cdot\) distributes over + in both arguments
- and 0 is a left and right annihilator, i.e., \(0 \cdot x = 0 = x \cdot 0\).

A semiring is **idempotent** if + is, i.e., if \(x + x = x\). Then + can be interpreted as (angelic) choice, with 0 modelling the most partial program with no transition possibilities at all, and \(\cdot\) as sequential composition, where 1 models the program skip. In this case, the relation \(x \leq y \iff x + y = y\) is a partial order, called the **natural order** on \(S\). It has 0 as its least element. Moreover, + and \(\cdot\) are isotone w.r.t. \(\leq\) and \(x + y\) is the least upper bound or join of \(x\) and \(y\) w.r.t. \(\leq\).

An idempotent semiring is **Boolean** if it also has a greatest lower bound or meet operation \(\land\), such that + and \(\land\) distribute over each other, and an operation \(\neg\) that satisfies de Morgan’s laws as well as \(x \land \neg x = 0\) and \(x + \neg x = \top\), where \(\top = \neg 0\) is the greatest element. In other words, a Boolean semiring is a Boolean algebra with a sequential composition operation. To save parentheses we use the convention that \(\land\) binds tighter than + but less tight than \(\cdot\) does. We use \(\land\) rather than \(\sqcap\) for the meet to avoid a clash of notation between semiring theory and the theory of UTP. To disambiguate the formulas we use a larger \(\land\) for meta-logical conjunction.

An important, even Boolean, semiring is \(\text{REL}(M) = \mathcal{P}(M \times M)\), the algebra of binary relations under union and composition over a set \(M\), of which the predicates of UTP form a special instance. The greatest element is \(\top = M \times M\).