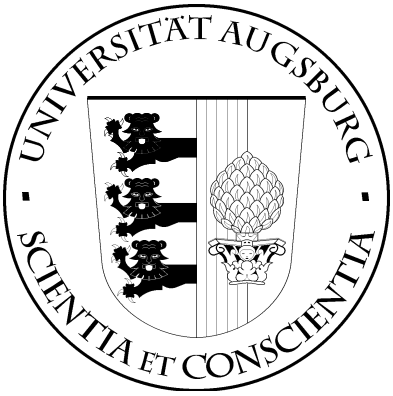
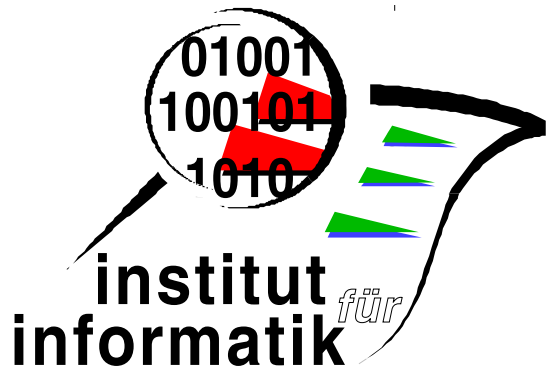
# Universität Augsburg




## Greedy-Like Algorithms in Kleene Algebra

**Bernhard Möller** **Georg Struth**

Report 2003-11 August 2003




Institut für Informatik
D-86135 Augsburg

# Greedy-Like Algorithms in Kleene Algebra*

Bernhard Möller Georg Struth

Institut für Informatik, Universität Augsburg
Universitätsstr. 14, D-86135 Augsburg, Germany
moeller,struth@informatik.uni-augsburg.de

**Abstract.** This paper provides an algebraic background for the formal derivation of greedy-like algorithms. Such derivations have previously been done in various frameworks including relation algebra. We propose Kleene algebra as a particularly simple alternative. Instead of converse and residuation we use modal operators that are definable in a wide class of algebras, based on domain/codomain or image/pre-image operations. By abstracting from earlier approaches we arrive at a very general theorem about the correctness of loops that covers particular forms of greedy algorithms as special cases.

**Keywords:** Idempotent semiring, Kleene algebra, image and preimage operation, modal operators, confluence, Geach formula, program development and analysis.

## 1 Introduction

This study is concerned with algebraic derivations and correctness proofs of greedy-like algorithms. These are algorithms that use a simple loop to calculate a global optimum. We present a fairly general correctness criterion and give sufficient criteria when iteration in a discrete partially ordered problem domain correctly implements the general algorithm scheme. Proper greedy algorithms are further specializions in which the loop steps are driven by an additional local optimality criterion. Earlier work on this latter subject has been performed using relation algebra (e.g. [1,2]) and other algebraic frameworks (e.g. [4]).

But why a re-development in Kleene algebra? The reason is that it allows a particularly simple, elegant and more general treatment. It avoids the concepts of converse and residual, as used in relation algebra, and exhibits the structure of the derivations more clearly. In particular, in a number of subderivations regular algebra instead of residual theory leads to more perspicuous and simple calculations. It also becomes clear that the theory of greedy algorithms is mainly about semi-commutation properties, e.g. that the local optimality criterion semi-commutes with the loop to achieve global optimality.

These semi-commutation properties are of the same nature as confluence properties in rewriting systems. To express them without converse we use modal

* Research partially sponsored by DFG Project InopSys — Interoperability of Calculi for System Modelling

operators as in dynamic logic. These can be added to Kleene algebra in a simple way. More generally, we use an algebra of predicate transformers (such as box and diamond) over a Kleene algebra which shows a Kleene algebra structure again. This algebra has many additional properties which should be useful for other applications as well.

The remainder of this paper is organized as follows. In Section 2 we recollect the notion of greediness and derive general conditions under which a greedy-like loop implements the specification of a corresponding algorithm. In Section 3 some of these conditions are established once and for all for a family of loops, among which there is a whole class of proper greedy algorithms. While so far everything has been worked out in relational algebra, Section 4 prepares the generalization to the converse-free framework of Kleene algebra by expressing the central type of conditions using the modal operators diamond and box. Section 5 then introduces Kleene algebra with domain and the abstract definition of the modalities. In Section 6 these are embedded into algebras of predicate transformers that allow a point-free treatment. Section 7 provides an algebraic equivalence proof of the (converse-free) modal Geach formula and the relational type of semi-commutation properties that are central to the correctness of our greedy-like algorithms. In Section 8 we can now replay the derivation of Sections 2 and 3 in Kleene algebra. Section 9 then reconstructs Curtis's classification of greedy algorithms in this more abstract and general setting; moreover, we are able to give completely point-free calculations. A brief Conclusion in Section 10 finishes the paper.

## 2 Looping for Optimality

Greedy algorithms are a special way of solving certain optimization problems. Their characteristics are that they proceed in a stepwise fashion without backtracking. At each step there is a set of choices from which a greedy algorithm always takes the one that seems best at the moment, i.e., it works locally without lookahead to the global optimum that is to be achieved eventually. Instances of this scheme are, e.g., shortest path and minimum spanning tree problems in graphs, the construction of Huffman codes and scheduling problems.

Of course, the greedy approach only works for certain types of problems, as is well-known from hiking in the mountains: always following the steepest path will lead to a local optimum, viz. the closest summit, but rarely to the highest summit of the whole area. The central principle that guarantees correctness of the greedy scheme is that *taking a local choice must not impair the possibility of reaching the global optimum.*

In this and the following section we derive, within the framework of relational algebra, general conditions under which a loop satisfies this principle. It turns out that local optimality is inessential; so we obtain a more general class of loops that we call *greedy-like.*

We start with a specification relation $T$ that connects inputs to admissible outputs and a relation $C$ that compares outputs and is meant to capture the notion of optimality. The derivation will exhibit our precise requirements on $C$.

A relation $R$ *improves* $T$ w.r.t. $C$ if it always relates inputs to outputs that are at least as good as those prescribed by $T$, in formulas

$$\forall\, x, y, z : xTy \,\wedge\, xRz \;\Rightarrow\; y\; C\; z,$$

which is equivalent to

$$T^{\smile}; R \;\subseteq\; C\ ,$$

where ; denotes usual relational composition and $T^{\smile}$ is the converse of $T$. Since $\emptyset$ trivially improves $T$, we are interested in the greatest improvement and define it by the Galois connection

$$X \;\subseteq\; \mathrm{GIMP}(T, C) \stackrel{\mathrm{def}}{\Leftrightarrow} T^{\smile}; X \;\subseteq\; C. \tag{1}$$

Using a residual, this could be expressed as $\mathrm{GIMP}(T, C) = T^{\smile} \backslash C$. However, we will not need any special properties of residuals.

We now want to calculate a sufficient criterion for a loop $\mathsf{while}\ P\ \mathsf{do}\ S \stackrel{\mathrm{def}}{=} (P\,;S)^*\,;\neg P$ to be such an improvement, i.e., to satisfy

$$\mathsf{while}\ P\ \mathsf{do}\ S \;\subseteq\; \mathrm{GIMP}(T, C)\ .$$

Here the loop condition $P$ is represented by a subidentity $P \;\subseteq\; I$, where $I$ is the identity relation.

Spelling out the definitions results in

$$T^{\smile}; (P\,;S)^*\,;\neg P \;\subseteq\; C.$$

We abstract a bit and try to answer the question when

$$U\,;V^*\,;Q \;\subseteq\; C \tag{2}$$

where additionally $Q \;\subseteq\; I$ is required.

Now, a standard result from regular algebra (see (14) in Section 5) is the semi-commutation property

$$W\,;X \;\subseteq\; Y\,;Z \;\Rightarrow\; W\,;X^* \;\subseteq\; Y^*\,;Z\ .$$

Hence (2) can be established given the following two conditions:

$$U\,;V \;\subseteq\; C\,;U\ , \tag{3}$$
$$U\,;Q \;\subseteq\; C\ , \tag{4}$$

since then

$$U\,;V^*\,;Q \;\subseteq\; C^*\,;U\,;Q \;\subseteq\; C^*\,;C \;=\; C^+\ .$$

If we now assume $C$ to be transitive, which is reasonable for a comparison relation, we have $C^+ \;\subseteq\; C$ and can draw the desired conclusion.

How can we, in turn, establish (3) and (4), at least in our special case? Translating back we get the proof obligations

$$T^\smile ; P ; S \ \subseteq \ C ; T^\smile, \tag{5}$$

$$T^\smile ; \neg P \ \subseteq \ C. \tag{6}$$

Let us interpret these conditions. (5) means that every pass through the loop body preserves the possibility of obtaining a solution that is at least as good as all possible solutions before. (6) means that upon loop termination no possible solution is better than the termination value.

An implementation of specification $T$ that always produces optimal solutions then is a relation that refines and improves $T$. So we define

$$\mathrm{OPT}(T, C) \stackrel{\text{def}}{=} T \cap \mathrm{GIMP}(T, C).$$

To achieve such an implementation we get the additional proof obligation

$$\mathsf{while}\ P\ \mathsf{do}\ S \ \subseteq \ T. \tag{7}$$

We will see how to deal with these three obligations in a special case for $T$.

## 3 Iterating Through the Problem Domain

We now want to decompose the specification relation $T$ into elementary steps $E$ from one element of the problem domain to another. Whereas we may start with arbitrary inputs as initial approximations, as outputs we admit only terminal elements from which no further elementary step is possible. Therefore we assume now that $T$ has the special shape

$$T \ = \ \mathsf{rep}\, E \ = \ E^* ; \neg \lceil E \ = \ \mathsf{while}\ \lceil E\ \mathsf{do}\ E \ , \tag{8}$$

where the *domain* of a relation $R$ is, as usual, defined by

$$\lceil R \stackrel{\text{def}}{=} R^\smile ; R \cap I \ . \tag{9}$$

Such a problem structure is found e.g. in matroids and greedoids [6,7] where it is additionally assumed that $T$ is a discrete strict-order and that all terminal (or maximal) elements, the *bases*, have the same height (also known as *rank* or *dimension*) in the associated Hasse diagram.

We try to calculate an implementation that traverses the problem domain without backtracking, i.e. using elementary steps only forward. This suggests trying $P ; S \ \subseteq \ E$. Now, by monotonicity of the star operation, proof obligation (7) can be fulfilled if additionally we can achieve $\neg P \ \subseteq \ \neg \lceil E$ or, equivalently, $\lceil E \ \subseteq \ P$. Sufficient conditions for these properties are

$$P ; S \ \subseteq \ E \qquad\qquad \lceil (P ; S) = \lceil E. \tag{10}$$

These are reasonable requirements, since they prevent that the iteration blocks at a non-terminal element.

Next, we deal with proof obligation (6), assuming (10). We calculate

$$
\begin{array}{ll}
& T^\smile ; \neg\ulcorner E \;\subseteq\; C \\
\Leftrightarrow & \quad \{\![ \text{ converse } ]\!\} \\
& \neg\ulcorner E \,;\, T \;\subseteq\; C^\smile \\
\Leftrightarrow & \quad \{\![ \text{ by (8) } ]\!\} \\
& \neg\ulcorner E \,;\, E^* \,;\, \neg\ulcorner E \;\subseteq\; C^\smile \\
\Leftrightarrow & \quad \{\![ \text{ unfold star } ]\!\} \\
& \neg\ulcorner E \,;\, (I \,\cup\, E \,;\, E^*) \,;\, \neg\ulcorner E \;\subseteq\; C^\smile \\
\Leftrightarrow & \quad \{\![ \text{ distributivity, } \neg\ulcorner E \,;\, E \,=\, \emptyset \text{ and idempotence of } \neg\ulcorner E \,]\!\} \\
& \neg\ulcorner E \;\subseteq\; C^\smile \\
\Leftarrow & \quad \{\![ \neg\ulcorner E \;\subseteq\; I \text{ and converse } ]\!\} \\
& I \;\subseteq\; C.
\end{array}
$$

So we can establish (6) provided $C$ is reflexive as well, i.e., a pre-order. This is again a reasonable requirement.

Proof obligation (5) cannot be simplified in such a general fashion; it is a generic condition that has to be considered individually in each case.

Our derivation can be summed up as follows.

**Theorem 3.1** *Suppose that $C$ is a pre-order and $T = \mathsf{rep}\, E$. If (5) and (10) hold then*

$$\mathsf{while}\ \ulcorner E\ \mathsf{do}\ S \;\subseteq\; \mathrm{OPT}(T, C)\ .$$

So far we still have a general scheme that does not specifically mention greediness. But we can refine $S$ further to choose in every step a locally optimal element. To this end we need yet another pre-order $L$ and stipulate

$$S \;\subseteq\; \mathrm{GIMP}(E, L)\ . \tag{11}$$

This now provides a truly greedy algorithm, the correctness of which is already shown by Theorem 3.1. It corresponds to Curtis's "Best-Global" algorithm [2].

## 4 From Converses to Modalities

The central step in the above derivation, viz. exhibiting conditions (5) and (6), uses only regular algebra. Hence it is an interesting question whether the derivation as a whole can be ported to Kleene algebra by eliminating the converse operation in some way. This would generalize the result to a much wider class of algebras.

In the above formulas the converse is used only in a very restricted way that reminds one of the relational formulation property of a general diamond (or confluence) property:

$$R^\smile ; S \;\subseteq\; T \,;\, U^\smile . \tag{12}$$

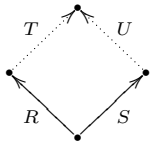
To bring (6) into this form, just compose the right hand side with $I^\smile$.

This observation is the key to success if one also remembers modal correspondence theory (see e.g. [12]), according to which the above formula is equivalent to both of the modal Geach formulas

$$\langle R\rangle[T]P \;\Rightarrow\; [S]\langle U\rangle P\;, \qquad\qquad \langle S\rangle[U]P \;\Rightarrow\; [R]\langle T\rangle P\;. \tag{13}$$

Now we are in good shape, since the modal operators $\langle\_\rangle$ and $[\_]$ can be defined as predicate transformers in Kleene algebra, cf. [3].

We shall use many instances of these formulas. Since one can easily confuse the rôles of the relations involved in the modal formulation, we shall illustrate these formulas by the following type of diagram:




When read as a confluence-type diagram, the solid arrows and their end points symbolize given elements and relations between them, whereas the dotted ones stand for a quantifier stipulating existence of a further element and appropriate relations to given elements. If one of the arrows is an identity, the diagram shrinks to a triangle.

## 5 Abstracting to Kleene Algebra

We now abstract from the relational setting and move into the more general framework of modal Kleene algebra.

First, a *Kleene algebra* (KA) [8] is a structure $(K,+,\cdot,^*,0,1)$ such that $(K,+,\cdot,0,1)$ is an idempotent semiring and $^*$ is a unary operation axiomatized by the equations and Horn sentences

$$1 + aa^* \le a^*, \tag{$*$-1}$$
$$1 + a^*a \le a^*, \tag{$*$-2}$$
$$b + ac \le c \Rightarrow a^*b \le c, \tag{$*$-3}$$
$$b + ca \le c \Rightarrow ba^* \le c, \tag{$*$-4}$$

for all $a,b,c \in K$. Here, $\le$ denotes the natural ordering on $K$ defined by $a \le b$ iff $a + b = b$. An important property that follows from these axioms is the semi-commutation law

$$ab \le cd \;\Rightarrow\; ab^* \le c^*d\;. \tag{14}$$

A special case of this establishes (2).

A KA is $^*$-*continuous* if for all $a,b,c$ we have

$$ab^*c \;=\; \sum_{i\in\mathbb{N}} ab^ic\;.$$

A *Kleene algebra with tests* (KAT) is a KA with a Boolean subalgebra $\mathsf{test}(K) \subseteq K$ in which 1 is the greatest element, 0 is the least element and $\cdot$ coincides with the meet operation. In a KAT we can again define the while loop as

$$\mathsf{while}\ p\ \mathsf{do}\ a \stackrel{\text{def}}{=} (pa)^*\neg p\ .$$

Finally, a *Kleene algebra with domain* (KAD) (see [3]) is a Kleene algebra with tests and an additional operation $\ulcorner\_ : K \to \mathsf{test}(K)$ such that for all $a, b \in K$ and $p, q \in \mathsf{test}(K)$,

$$a \leq \ulcorner(a)a\ , \tag{d1}$$
$$\ulcorner(pa) \leq p\ , \tag{d2}$$
$$\ulcorner(ab) = \ulcorner(a\ulcorner b)\ . \tag{d3}$$

Let us explain these axioms. As in the algebra of relations, multiplication with a test from the left or right means domain or range restriction, resp. Now first, since $\ulcorner(a) \leq 1$ by $\ulcorner(a) \in \mathsf{test}(K)$, monotonicity of multiplication shows that (d1) can be strengthened to an equality expressing that domain restriction to the full domain is no restriction at all. (d2) means that after restriction the remaining domain must satisfy the restricting test. The axiom (d3) serves to make the modal operators below well-behaved w.r.t. composition. An important consequence of the axioms is that $\ulcorner\_$ preserves arbitrary existing suprema [11].

Examples of KADs are the algebra of concrete relations, where $\ulcorner\_$ coincides with the operation defined in (9), the algebra of path sets in a directed graph (see e.g. [10]) and Kleene's original algebra of formal languages.

In a KAD, the (forward) modal operators *diamond* and *box* can be defined by

$$\langle a\rangle p \stackrel{\text{def}}{=} \ulcorner(ap)\ , \qquad [a]p \stackrel{\text{def}}{=} \neg\langle a\rangle\neg p\ .$$

This definition is adequate, since it makes the diamond coincide with the inverse image operator. Axiom (d3) implies

$$\langle ab\rangle p = \langle a\rangle\langle b\rangle p\ . \tag{15}$$

The modal operators for a test $q$ are given by

$$\langle q\rangle p = qp\ , \qquad [q]p = q \to p \stackrel{\text{def}}{=} \neg q + p\ . \tag{16}$$

For the above-mentioned connection between confluence-type formulas and the Geach formula we introduce the following notion. A KAD *with converse* is a KAD $K$ with an additional operation $\_^\smile : K \to K$ that is an involution, distributes over $+$, is the identity on tests and is contravariant over $\cdot$, i.e., satisfies $(ab)^\smile = b^\smile a^\smile$.

One can show (see again [3]) that over a KAD with converse the axioms (d1) and (d2) imply the Galois connection

$$\langle a^\smile\rangle p \leq q \Leftrightarrow p \leq [a]q\ . \tag{17}$$

It follows that all predicate transformers $\lambda p \,.\, \langle a \rangle p$ are universally disjunctive, i.e., preserve arbitrary existing suprema. The latter results generalizes to KADs that also provide a codomain operation, since there one can also define the backward modal operators and replace $\langle a^\smile \rangle$ by the backward diamond of $a$.

Moreover, from (16) we get, for predicates $p, q, r$, even in KADs without converse,

$$p \leq [r]q \;\Leftrightarrow\; \langle r \rangle p \leq q \;. \tag{18}$$

## 6 Predicate Transformers

Our previous relation-algebraic derivation can now be mimicked more abstractly at the level of predicate transformers over an arbitrary KAD. In particular, we do not need the carrier of the algebra to be a complete Boolean algebra as in the case of relation algebra; also, residuals and converse can be avoided.

Assume a t-semiring $(K, +, \cdot, 0, 1)$. By a *predicate transformer* we mean a function $f : \mathsf{test}(K) \to \mathsf{test}(K)$. It is *disjunctive* if $f(p + q) = f(p) + f(q)$ and *conjunctive* if $f(p \cdot q) = f(p) \cdot f(q)$. It is *strict* if $f(0) = 0$. Finally, $id$ is the identity transformer and $\circ$ denotes function composition.

Let $P$ be the set of *all* predicate transformers, $M$ the set of monotonic and $D$ the set of strict and disjunctive ones. Under the pointwise ordering $f \leq g \stackrel{\text{def}}{\Leftrightarrow} \forall\, p \,.\, f(p) \leq g(p)$, $P$ forms a lattice where the supremum $f \oplus g$ and infimum $f \odot g$ of $f$ and $g$ are the pointwise liftings of $+$ and $\cdot$, resp.:

$$(f \oplus g)(p) \stackrel{\text{def}}{=} f(p) + g(p), \qquad (f \odot g)(p) \stackrel{\text{def}}{=} f(p) \cdot g(p).$$

The least element of $P$ (and $D$) is the constant 0-valued function $\mathbf{0}(p)$.

The structure $(P, \oplus, \circ, , id)$ is an idempotent left semiring which means that $\mathbf{0}$ is only a left annihilator and multiplication is distributive over addition in its left argument only. In fact, in its left argument $\circ$ even preserves arbitrary existing suprema, as the following calculation shows:

$$((\Sigma f_i) \circ g)(x) \;=\; (\Sigma f_i)(g(x)) \;=\; \Sigma f_i(g(x)) \;=\; \Sigma(f_i \circ g)(x) \;.$$

The substructure $(D, \oplus, \circ, \mathbf{0}, id)$ is an idempotent semiring. The modal operator $\langle \_ \rangle$ provides a left-semiring homomorphism.

If $\mathsf{test}(K)$ is a complete Boolean algebra then $P$ is a complete lattice with $D$ as a complete sublattice. Hence we can extend $P$ and $D$ by two star operations using the standard least fixpoint definitions:

$$f^* \stackrel{\text{def}}{=} \mu g \,.\, id \oplus g \circ f \qquad {}^*f \stackrel{\text{def}}{=} \mu g \,.\, id \oplus f \circ g \;,$$

where $\mu$ is the least-fixpoint operator. Under ${}^*$-continuity one has

$$f^* \;=\; \sum_{i \in \mathbb{N}} f^i \tag{19}$$

and hence

$$f^* \leq {}^*f \ , \tag{20}$$

since an easy induction shows $a^i \leq^* f$ for all $i \in \mathbb{N}$.

The converse inequation does not hold in $P$, but in $D$ it does, since $D$ is a KA. Hence both operators coincide in $D$. In $P$, the right star $f^*$ is more pleasant to work with because of the asymmetry in distributivity. A further property is

$$h \text{ universally disjunctive} \wedge h \circ f \leq f \circ h \Rightarrow h \circ f^* \leq f^* \circ h. \tag{21}$$

The structure $(D, \oplus, \circ, \mathbf{0}, id, ^*)$ is a KA, the *predicate transformer algebra*. However, we will work in the encompassing $(P, \oplus, \circ, \mathbf{0}, id, ^*)$ which might be called a left KA.

## 7 Properties of Modal Operators

We now concentrate on the modal predicate transformers $\langle \_ \rangle$ and $[\_]$. Since they are functions, they satisfy the principle of extensional inequality, i.e., for $a, b \in K$,

$$\langle a \rangle \leq \langle b \rangle \ \Leftrightarrow \ \forall \, p \, . \, \langle a \rangle p \leq \langle b \rangle p \ .$$

For the rest of the paper we will work as much as possible at this point-free level of operator algebras. To smoothen the notation, we will denote composition of predicate transformers by mere juxtaposition.

First, we note that in a KAD with converse the predicate-level Galois connection (17) implies the predicate-transformer-level cancellation laws

$$\langle a^\smile \rangle [a] \leq \langle 1 \rangle \leq [a] \langle a^\smile \rangle \ . \tag{22}$$

Moreover, (17) lifts to a Galois connection between predicate transformers:

$$f \leq [a] g \ \Leftrightarrow \ \langle a^\smile \rangle f \leq g \ . \tag{23}$$

Next, we note

**Lemma 7.1** *If the underlying* KA *is* $^*$*-continuous, then*

$$\langle a^* \rangle \ = \ \langle a \rangle^* \tag{24}$$

*Hence, in this case* $\langle \_ \rangle$ *is a homomorphism between left KAs.*

*Proof.* We calculate

$$\begin{aligned} \langle a^* \rangle p = \ulcorner((\Sigma a^i)p) \ &= \ \ulcorner(\Sigma a^i p) \ = \ \Sigma \ulcorner(a^i p) \\ &= \Sigma \langle a^i \rangle p \ = \ \Sigma \langle a \rangle^i p \ = \ (\Sigma \langle a \rangle^i) p \ = \ \langle a \rangle^* p \ . \end{aligned}$$

We have used, in this order, the definition of $\langle \_ \rangle$, $^*$-continuity twice, that $\ulcorner\_$ preserves arbitrary existing suprema, the definition of $\langle \_ \rangle$, that $\langle \_ \rangle$ distributes through $\cdot$ (equation ( 15)), the definition of suprema in $P$ and (19). □

We will now give an abstract proof of equivalence of the Geach formula (13) and the confluence property (12). We do this in KADs that are *extensional* (or *separable* as they are called in dynamic algebra), i.e., satisfy

$$a \leq b \Leftrightarrow \langle a \rangle \leq \langle b \rangle \ . \tag{25}$$

Note that only the direction from right to left must be required, the other one holds in KAD by monotonicity.

Moreover, we use an extension of KAT by a *converse* operation $\_^\smile$ that is an involution, distributes through $+$ and is contravariant over $\cdot$, i.e., satisfies

$$(ab)^\smile = b^\smile a^\smile ,$$

and preserves predicates, i.e., $p^\smile = p$.

Now we can show

**Theorem 7.2** *In an extensional* KAD *with converse,*

$$a^\smile b \leq c d^\smile \Leftrightarrow \langle b \rangle [d] \leq [a] \langle c \rangle . \tag{26}$$

*Proof.* ($\Rightarrow$) We calculate

$$\begin{aligned} a^\smile b \leq c d^\smile &\Leftrightarrow \langle a^\smile b \rangle \leq \langle c d^\smile \rangle \\ &\Leftrightarrow \langle a^\smile \rangle \langle b \rangle \leq \langle c \rangle \langle d^\smile \rangle \\ &\Leftrightarrow \langle b \rangle \leq [a] \langle c \rangle \langle d^\smile \rangle \\ &\Rightarrow \langle b \rangle [d] \leq [a] \langle c \rangle \langle d^\smile \rangle [d] \\ &\Rightarrow \langle b \rangle [d] \leq [a] \langle c \rangle . \end{aligned}$$

The first step uses (25), the second step locality, the third step (17), the fourth step monotonicity, the fifth step (22).

($\Leftarrow$) Let $\langle b \rangle [d] \leq [a] \langle c \rangle$. Then

$$\langle b \rangle \leq \langle b \rangle [d] \langle d^\smile \rangle \leq [a] \langle c \rangle \langle d^\smile \rangle \ .$$

Then the proof continues like for ($\Rightarrow$), read upside down. □

A special case of the Geach formula deals with the domain operator. Its relational definition (9) implies $\ulcorner R \ \subseteq \ R \,;\, R^\smile$, admitting a certain relaxation of domain constraints in our derivations. Using the Geach formula this translates into the modal Kleene formula

$$\langle \ulcorner a \rangle [a] \leq \langle a \rangle \ , \tag{27}$$

which can even be shown to hold in *all* KADs, not only in extensional KADs with converse.

## 8 Looping for Optimality in Kleene Algebra

Using the Geach formula we can now replay our previous derivation in the more abstract setting of KAs. Specifications and implementations are now simply elements of a KA.

A difference to the relational setting is that we cannot carry over GIMP directly, since in general KAs residuals need not exist.

But for our derivation there is no need to internalize the concept of greatest improvement; rather we use a characterizing predicate in which the right hand side of (1) is replaced by the corresponding modal formula and $c$ now plays the rôle of $C$:

$$\mathrm{IMP}(x,t,c) \stackrel{\mathrm{def}}{\Leftrightarrow} \langle x\rangle \leq [t]\langle c\rangle. \tag{28}$$

Now we need to find a sufficient criterion for $\mathrm{IMP}(\mathsf{while}\ p\ \mathsf{do}\ s,t,c)$ which spells out to $\langle (ps)\neg p\rangle \leq [t]\langle c\rangle$. As in Section 2, we abstract and want to achieve $\langle v^*\, q\rangle \leq [t]\langle c\rangle$ which by (24) is equivalent to $\langle v\rangle^*\langle q\rangle \leq [t]\langle c\rangle$. Since by (20) $\langle v\rangle^* \leq {}^*\langle v\rangle$, it suffices to show ${}^*\langle v\rangle\langle q\rangle \leq [t]\langle c\rangle$. By the least-fixpoint property of the left star this is implied by $\langle q\rangle \oplus \langle v\rangle[t]\langle c\rangle \leq [t]\langle c\rangle$, equivalently

$$\langle q\rangle \leq [t]\langle c\rangle \ \ \wedge\ \ \langle v\rangle[t]\langle c\rangle \leq [t]\langle c\rangle\ . \tag{29}$$

The second conjunct, in turn, is implied by

$$\langle v\rangle[t] \leq [t]\langle c\rangle \tag{30}$$

provided $cc \leq c$.

The full specification of our task in KA reads, analogously to Section 2,

$$\mathrm{KAOPT}(x,t,c) \stackrel{\mathrm{def}}{=} x \leq t \wedge \mathrm{IMP}(x,t,c).$$

This yields the additional proof obligation

$$\mathsf{while}\ p\ \mathsf{do}\ s \leq t. \tag{31}$$

Assume now, as in (8), that $t = \mathsf{rep}\, e = e^*\,;\neg\ulcorner e$.

With the same derivation as in Section 3 we can show that the following property is sufficient for (31):

$$ps \leq e \wedge \ulcorner(ps) = \ulcorner e. \tag{32}$$

Next we note that (30) and the first conjunct of (29) spell out to

$$\langle ps\rangle[t] \leq [t]\langle c\rangle, \tag{33}$$

$$\langle \neg p\rangle \leq [t]\langle c\rangle. \tag{34}$$

Again, (34) is implied by (32) if $c$ is reflexive.

Summing up, we have the following KA variant of Theorem 3.1:

**Theorem 8.1** *Suppose that $c$ is a pre-order and $t = \mathsf{rep}\, e$. If (32) and (33) hold then*

$$\mathrm{KAOPT}(\mathsf{while}\ \ulcorner e\ \mathsf{do}\ s,t,c)\ .$$

## 9 Classifying Greedy Algorithms

In the present section we demonstrate that the modal approach does indeed provide a convenient tool for many further applications. We demonstrate this in an abstract reconstruction of Curtis's classification of Greedy algorithms in [2] to which we also refer the reader for concrete examples of the various types of algorithms. The modal operators again lead to considerably more concise proofs than the original relational ones.

Throughout this section we assume the following. First, $t$ is the specification and $c$ and $l$ are pre-orders that model global and local comparison, respectively. Second, $t = \mathsf{rep}\, e$ is the specification that completes initial approximative solutions to terminal ones using elementary steps $e$. Third, $g \leq e$ is supposed to be a greedy step that satisfies, analogously to (10) and (11),

$$\ulcorner g \;=\; \ulcorner e \;, \tag{35}$$

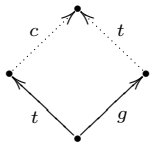
$$\mathrm{IMP}(g,e,l)\,,\;\; \text{i.e.,}\;\; \langle g\rangle \leq [e]\langle l\rangle \qquad (\Leftrightarrow\; e^{\smile} g \leq l)\;. \tag{36}$$

In the following theorems we will always list the conditions both in modal notation as obtained by the Geach formula (26) and in the one of KADs with converse and illustrate them by diagrams.

Immediately from Theorem 8.1 we obtain the following description of the first class of greedy algorithms:

**Theorem 9.1 (Best-Global)** $\mathrm{KAOPT}(\mathsf{rep}\, g, t, c)$ follows from

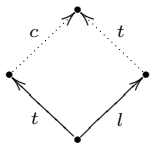
$$\langle g\rangle[t] \leq [t]\langle c\rangle \qquad (\Leftrightarrow\; t^{\smile} g \leq c t^{\smile})\;. \tag{37}$$




The next class is characterized by

**Theorem 9.2 (Better-Global)** $\mathrm{KAOPT}(\mathsf{rep}\, g, t, c)$ follows from

$$\langle l\rangle[t] \leq [t]\langle c\rangle \qquad (\Leftrightarrow\; t^{\smile} l \leq c t^{\smile})\;. \tag{38}$$




This condition says that for any pair of local choices the locally better one has a completion at least as good as any completion of the locally worse one.

*Proof.* We show that the assumptions imply condition (37) of Theorem 9.1. We calculate

$$\langle g\rangle[t] \le [e]\langle l\rangle[t] \le [e][t]\langle c\rangle = [\ulcorner e][t]\langle c\rangle \ .$$

The first step uses (36), the second one (38) and the third one the definition of $t = \mathsf{rep}\, e$. But by (18) and (35) this is equivalent to the claim. □

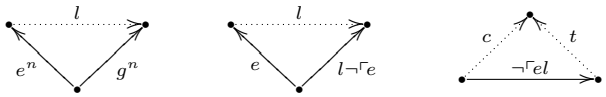
The third class of greedy algorithms has a more elaborate set of preconditions.

**Theorem 9.3 (Best-Local)** If we assume $^*$-continuity, KAOPT($\mathsf{rep}\, g, t, c$) follows from

$$\forall\, n \in \mathbb{N} : \langle g^n\rangle \le [e^n]\langle l\rangle \ \ (\Leftrightarrow\ \forall\, n \in \mathbb{N} : (e^n)^\smile g^n \le l)\ , \tag{39}$$

$$\langle l\rangle\langle \neg\ulcorner e\rangle \le [e]\langle l\rangle \ \ (\Leftrightarrow\ e^\smile l \neg\ulcorner e \le l)\ , \tag{40}$$

$$\langle l\rangle[t] \le [\neg\ulcorner e]\langle c\rangle \ \ (\Leftrightarrow\ \neg\ulcorner e l \le c t^\smile)\ . \tag{41}$$




Here the local choice is made depending on the history of choices before. The first of these conditions says that each step produces an approximation to the final optimum that is optimal among the approximations that can be obtained with he same number of steps. The other two conditions state that once the sequence of greedy steps finishes, completions of other approximations cannot improve the result any more.

*Proof.* First we note that, by an easy induction using idempotence of tests, in particular $\langle \neg\ulcorner e\rangle = \langle \neg\ulcorner e\rangle\langle \neg\ulcorner e\rangle$, condition (40) generalizes to

$$n \ge 0 \ \Rightarrow\ \langle l\rangle\langle \neg\ulcorner e\rangle \le [e^n]\langle l\rangle\ . \tag{42}$$

The proof proper is performed by showing that the assumptions imply condition (37) of Theorem 9.1, i.e., $\langle g\rangle[t] \le [t]\langle c\rangle$.

Using $^*$-continuity and $[a+b] = [a] \odot [b]$ this reduces to

$$\forall\, n \in \mathbb{N} : \langle g\rangle[t] \le [e^n][\neg\ulcorner e]\langle c\rangle\ .$$

For $n = 0$, we use idempotence of predicates and (18) to see that

$$\langle g\rangle[t] \le [\neg\ulcorner e]\langle c\rangle \ \Leftrightarrow\ \langle \neg\ulcorner e\rangle\langle g\rangle[t] \le [\neg\ulcorner e]\langle c\rangle\ .$$

Now we calculate, using (35),

$$\langle \neg\ulcorner e\rangle\langle g\rangle[t] \ =\ \langle \neg\ulcorner g\rangle\langle g\rangle[t] \ =\ \langle \neg\ulcorner gg\rangle[t] \ =\ \langle 0\rangle[t] \ =\ \langle 0\rangle\ ,$$

and the claim is shown.

For fixed $n > 0$ we split the greedy step $g$ into the part $g\,\ulcorner(g^{n-1})$ that admits at least $n-1$ further greedy steps, and its relative complement $g\,\neg\ulcorner(g^{n-1})$, and show

separately $\langle g\,\ulcorner(g^{n-1})\rangle[t] \leq r$ and $\langle g\,\neg\ulcorner(g^{n-1})\rangle[t] \leq r$, where $r \stackrel{\text{def}}{=} [e^n][\neg\ulcorner e]\langle c\rangle$. For the first part we calculate

$$\langle g\rangle\langle\ulcorner(g^{n-1})\rangle[t] \leq \langle g\rangle\langle\ulcorner(g^{n-1})\rangle[g^{n-1}][t] \leq \langle g\rangle\langle g^{n-1}\rangle[t] \leq [e^n]\langle l\rangle[t] \leq [e^n][\neg\ulcorner e]\langle c\rangle \ .$$

The first step uses $g^{n-1}t \leq t$ and antitonicity of $[\_]$. The second step follows by (27). The third step joins powers and uses (39). The final step employs (41).

For the second part we want to use again (27) and so have to replace $\neg\ulcorner(g^{n-1})$ by a positive domain expression. We calculate, for arbitrary $i$,

$$\neg\ulcorner(g^i) \ = \ \neg(\ulcorner(g^i\,\ulcorner g) + \ulcorner(g^i\,\neg\ulcorner g)) \ = \ \neg\ulcorner(g^i\,\ulcorner g)\neg\ulcorner(g^i\,\neg\ulcorner g) \ = \ \neg\ulcorner(g^{i+1})\neg\ulcorner(g^i\,\neg\ulcorner g) \ .$$

Using only the $\geq$ half of this equality and shunting we obtain

$$\neg\ulcorner(g^{i+1}) \leq \neg\ulcorner(g^i) + \ulcorner(g^i\,\neg\ulcorner g) \ ,$$

and an easy induction shows $\neg\ulcorner(g^n) \leq \Sigma_{i<n}\ulcorner(g^i\,\neg\ulcorner g)$. By disjunctivity of $\langle\_\rangle$ our claim is thus established if $\langle g\rangle\langle\ulcorner(g^m\,\neg\ulcorner g)\rangle[t] \leq r$ for all $m < n$. We calculate

$$\begin{array}{ll}
 & \langle g\rangle\langle\ulcorner(g^m\,\neg\ulcorner g)\rangle[t] \\
\leq & \quad \{\![\ g^m\,\neg\ulcorner g t \leq t \text{ and antitonicity of } [_]\ ]\!\} \\
 & \langle g\rangle\langle\ulcorner(g^m\,\neg\ulcorner g)\rangle[g^m\neg\ulcorner g][t] \\
\leq & \quad \{\![\ \text{by (27)}\ ]\!\} \\
 & \langle g\rangle\langle g^m\,\neg\ulcorner g\rangle[t] \\
= & \quad \{\![\ \text{powers and (35)}\ ]\!\} \\
 & \langle g^{m+1}\rangle\langle\neg\ulcorner e\rangle[t] \\
\leq & \quad \{\![\ \text{by condition (39)}\ ]\!\} \\
 & [e^{m+1}]\langle l\rangle\langle\neg\ulcorner e\rangle[t] \\
\leq & \quad \{\![\ \text{by (42) and } n > m+1\ ]\!\} \\
 & [e^{m+1}][e^{n-m-1}]\langle l\rangle[t] \\
\leq & \quad \{\![\ \text{joining powers and using condition (41)}\ ]\!\} \\
 & [e^n][\neg\ulcorner e]\langle l\rangle \ .
\end{array}$$

□

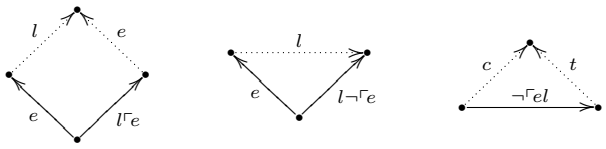
The final class of algorithms is given by

**Theorem 9.4 (Better-Local)** Under $^*$-continuity KAOPT($\mathsf{rep}\,g, t, c$) follows from

$$\langle l\rangle\langle\ulcorner e\rangle[e] \leq [e]\langle l\rangle \ \ (\Leftrightarrow\ e^\smile l\,\ulcorner e \leq l e^\smile) \ , \tag{43}$$

$$\langle l\rangle\langle\neg\ulcorner e\rangle \leq [e]\langle l\rangle \ \ (\Leftrightarrow\ e^\smile l\,\neg\ulcorner e \leq l) \ , \tag{44}$$

$$\langle l\rangle[t] \leq [\neg\ulcorner e]\langle c\rangle \ \ (\Leftrightarrow\ \neg\ulcorner e\, l \leq c t^\smile) \ . \tag{45}$$



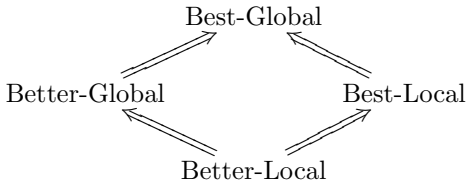

This essentially says that for any two local choices and any one-step extension of the locally worse one there is a locally better one-step extension of the locally better one.

*Proof.* We show that condition (38) of Theorem 9.2 is satisfied. First, using antitonicity of $[\_]$ and distributivity, we obtain

$$\begin{aligned}&(43) \wedge (44) \;\Leftrightarrow\; \langle l\rangle(\langle\ulcorner e\rangle[e] \oplus \langle\neg\ulcorner e\rangle[1]) \leq [e]\langle l\rangle \;\Rightarrow\\ &\langle l\rangle(\langle\ulcorner e\rangle[e+1] \oplus \langle\neg\ulcorner e\rangle[e+1]) \leq [e]\langle l\rangle \;\Leftrightarrow\; \langle l\rangle[e+1] \leq [e]\langle l\rangle\ .\end{aligned}$$

Now dualization of (14) together with the equivalence in (13) allows us to infer $\langle l\rangle[e^*] = \langle l\rangle[(e+1)^*] \leq [e^*]\langle l\rangle$, from which by condition (45) we get $\langle l\rangle[t] = \langle l\rangle[e^*][t] \leq [e^*]\langle l\rangle[t] \leq [e^*][\neg\ulcorner e]\langle c\rangle = [t]\langle c\rangle$. □

Curtis's classification is completed by showing the following relationship between the algorithm classes:




Except for the relation between Better-Local and Best-Local this was was established by the proofs of the previous Theorems. So we add

**Theorem 9.5** *The Better-Local conditions imply the Best-Local conditions.*

*Proof.* It suffices to show (39). This is done by induction on $n$. For $n = 0$ the claim follows from $1 \leq l$. For the induction step we calculate

$$\begin{aligned}\langle g^{n+1}\rangle \leq [e^n]\langle l\rangle\langle g\rangle \leq [e^n]\langle l\rangle\langle\ulcorner g\rangle\langle g\rangle \leq\\ [e^n]\langle l\rangle\langle\ulcorner g\rangle[e]\langle l\rangle \leq [e^n][e]\langle l\rangle\langle l\rangle \leq [e^{n+1}]\langle l\rangle\ .\end{aligned}$$

The first step splits a power and uses the induction hypothesis. The second step uses a domain law. The third step employs (36). The fourth step uses (35) and (43). The last step joins powers and uses transitivity of $l$. □

## 10 Conclusion

We have shown that a concise algebraic derivation of a general greedy-like algorithm can be obtained in the framework of Kleene algebra. The more pristine

framework avoids detours through residuals and leads to a simpler correctness proof than in [2,4].

The treatment has exhibited an interesting relation with semi-commutation properties as known from rewriting and allegories [5]. The connection to KA has already been explored in [13].

Doing away with converse has led us into the interesting and very well-behaved algebra of predicate transformers. In it we can express properties such as $\langle a^* \rangle = \langle a \rangle^*$ that cannot even be formulated in dynamic logic. We are therefore convinced that this algebra will have many further applications.

**Acknowledgement** We are grateful to S. Curtis for an enlightening discussion on greedy algorithms. Helpful remarks were provided by R. Backhouse and T. Ehm.

## References

1. R.S. Bird, O de Moor: Algebra of programming. Prentice Hall 1997
2. S.A. Curtis: A relational approach to optimization problems. D.Phil. Thesis. Technical Monograph PRG-122, Oxford University Computing Laboratory 1996
3. J. Desharnais, B. Möller, and G. Struth: Kleene algebra with domain. Technical Report 2003-07, Universität Augsburg, Institut für Informatik, 2003
4. J.E. Durán: Transformational derivation of greedy network algorithms from descriptive specifications. In: E.A. Boiten, B. Möller (eds.): Mathematics of program construction. Lecture Notes in Computer Science **2386**. Springer 2002, 40–67
5. P. Freyd, A. Scedrov: Categories, allegories. North-Holland 1990
6. P. Helman, B.M.E. Moret, H.D. Shapiro: An Exact Characterization of Greedy Structures. SIAM Journal on Discrete Mathematics **6**, 274-283 (1993)
7. B. Korte, L. Lovász, R. Schrader: Greedoids. Heidelberg: Springer 1991
8. D. Kozen:A completeness theorem for Kleene algebras and the algebra of regular events. Information and Computation **110**, 366–390 (1994)
9. D. Kozen: Kleene algebra with tests. Transactions on Programming Languages and Systems **19**, 427–443 (1997)
10. B. Möller: Derivation of graph and pointer algorithms. In: B. Möller, H.A. Partsch, S.A. Schuman (eds.): Formal program development. Lecture Notes in computer science **755**. Springer 1993, 123–160
11. B. Möller and G. Struth: Modal Kleene algebra and partial correctness. Technical Report 2003-08, Universität Augsburg, Institut für Informatik, 2003
12. S. Popkorn: First steps in modal logic. Cambridge University Press 1994
13. G. Struth: Calculating Church-Rosser proofs in Kleene algebra. In: H.C.M. de Swart (ed.): Relational Methods in Computer Science, 6th International Conference. Lecture Notes in Computer Science **2561**. Springer 2002, 276–290