The Axiomatization of Overriding and Update[☆]

Jasper Berendsen, David N. Jansen, Julien Schmaltz, Frits W. Vaandrager[*]

Institute for Computing and Information Sciences
Radboud University Nijmegen, The Netherlands

Abstract

There are only very few natural ways in which arbitrary functions can be combined. One composition operator is *override*: for arbitrary functions f and g, $f \rhd g$ is the function with domain $\mathsf{dom}(f) \cup \mathsf{dom}(g)$ that behaves like f on $\mathsf{dom}(f)$ and like g on $\mathsf{dom}(g) \setminus \mathsf{dom}(f)$. Another operator is *update*: $f[g]$ has the same domain as f, behaves like f on $\mathsf{dom}(f) \setminus \mathsf{dom}(g)$, and like g on $\mathsf{dom}(f) \cap \mathsf{dom}(g)$. These operators are widely used, especially within computer science, where for instance $f[g]$ may denote the new state that results when in state f the updates given as g are applied. It is therefore surprising that thus far no axiomatization of these operators has been proposed in the literature. As an auxiliary operator we consider the *minus* operator: $f - g$ is the restriction of f to the domain $\mathsf{dom}(f) \setminus \mathsf{dom}(g)$. The update operator can be defined in terms of override and minus. We present five equations that together constitute a sound and complete axiomatization of override and minus. As part of our completeness proof, we infer a large number of useful derived laws using the proof assistant Isabelle. With the help of the SMT solver Yices, we establish independence of the axioms. Thus, our axiomatization is also minimal. Finally, we establish that override and minus are functionally complete in the sense that any operation on general functions that corresponds to a valid coloring of a Venn diagram can be described using just these two operations.

Keywords: Overriding, Function algebra, Equational logic, Completeness, SMT solvers, Automated deduction
2010 MSC: 08A02 Relational systems, laws of composition, 03C05 Equational classes, universal algebra, 03B35 Mechanization of proofs and logical operations, 08-04 General algebraic systems: Explicit machine computation and programs

1. Introduction

There are only very few natural ways in which arbitrary functions can be combined. One example is the *override* operator $\rhd$. For arbitrary functions f and g, $f \rhd g$ is the combined function where f overrides g for all elements in the intersection of their domains. That is, $f \rhd g$ is the function with domain $\mathsf{dom}(f) \cup \mathsf{dom}(g)$ satisfying, for all $v \in \mathsf{dom}(f) \cup \mathsf{dom}(g)$,

$$(f \rhd g)(v) \triangleq \begin{cases} f(v) & \text{if } v \in \mathsf{dom}(f) \\ g(v) & \text{if } v \in \mathsf{dom}(g) \setminus \mathsf{dom}(f). \end{cases}$$

Essentially, this is the overriding operator “⊕” from Z [14]. On finite domains, the operator is also defined in VDM [10], where it is written †. We prefer not to use a symmetric symbol for an asymmetric (non-commutative) operator. There appears to be no commonly accepted name or symbol to denote this important operator. It is introduced on an ad hoc basis in many papers, and properties of the operator are used, implicitly or explicitly, in numerous proofs.

[☆] Research supported by NWO/EW project 612.000.103 Fault-tolerant Real-time Algorithms Analyzed Incrementally (FRAAI). The research leading to these results has received funding from the European Community’s Seventh Framework Programme under grant agreement no 214755 (QUASIMODO).

[*] Corresponding author.
Email address: F.Vaandrager@cs.ru.nl (Frits W. Vaandrager)

A related composition operator is *update:* $f[g]$ has the same domain as f, behaves like f on $\mathsf{dom}(f) \setminus \mathsf{dom}(g)$, and like g on $\mathsf{dom}(f) \cap \mathsf{dom}(g)$. Thus, if $h \lceil X$ restricts the domain of a function h to X, we can define the update operator by $f[g] \triangleq (g \rhd f) \lceil \mathsf{dom}(f)$. The update operator is also widely used, especially within computer science. One example is the denotational semantics of assignment [1, 16], where $f[g]$ denotes the new state that results when in state f the updates given as g are applied. Another example is the semantics of Statecharts as described in several publications, e. g. [6]. It uses the update operator $\cdot[\cdot]$ to denote assignments (multiple assignments may occur simultaneously), and circumscribes the update of a state configuration by set operations. As a final example we mention the work of Goldberg and Tarjan [8], where a sequence of functions is considered whose limit is a flow function. Every function is generated from the previous one by updating a number of function values. The update is denoted like a (simultaneous) assignment to some of the function values.

It is surprising that thus far no axiomatization of override and update has been proposed in the literature. Several authors observe that override is associative and idempotent [10, 14]. Various papers mention laws such as $f[g][g] = f[g]$. In our work on compositional abstraction for timed automata [3], we stated a number of laws for override and update as part of a proof of associativity of a Uppaal style parallel composition operator. Nevertheless, to the best of our knowledge, completeness has thus far not been studied.

The main result of the present paper is a sound and complete equational axiomatization of override and update. In our axiomatization we use the auxiliary operator *minus:* $f - g$ is the restriction of f to the domain $\mathsf{dom}(f) \setminus \mathsf{dom}(g)$. That is, $f - g = f \lceil (\mathsf{dom}(f) \setminus \mathsf{dom}(g))$. The update operator is then defined in terms of override and minus. Our axiomatization is also minimal, in the sense that all axioms are independent. It is not possible to prove any of the axioms from the remaining ones. Finally, we show that the override and minus operations are functionally complete in the sense that any operation on general functions that corresponds to a valid coloring of a Venn diagram can be described using just these two operations.

In Section 2, we present five equations and show that together they constitute a sound axiomatization of override and minus. In Section 3, we infer a large number of derived laws using the proof assistant Isabelle. These laws are heavily used in Section 4, where we establish completeness of the five equations. In Section 5, we prove minimality of the axioms with the help of the SMT solver Yices. Section 6 discusses the functional completeness of override and minus. Finally, Section 7 presents our conclusions and open questions.

2. The Axioms

We consider the signature Σ consisting of two binary (infix) operator symbols $\rhd$ and $-$, and the constant symbol $\emptyset$. We refer to $\rhd$ as the *override* symbol, $-$ as the *minus* symbol, and $\emptyset$ as *empty*. Using elements from this signature and *variables* $x, y, z, \ldots$ we build *terms* and *equations* in the standard way. We use $t, u, \ldots$ to denote terms over signature Σ, and $t \equiv u$ to denote syntactic equality of terms t and u. Figure 1 introduces a set E of five equations over the signature.

$$\begin{array}{rrcll}
\text{idempotence} & x \rhd x & = & x & (1) \\
\text{empty} & x - x & = & \emptyset & (2) \\
\text{weak commutativity} & x \rhd y & = & (y - x) \rhd x & (3) \\
\text{double minus} & x - (y - z) & = & (x - y) \rhd (x - (x - z)) & (4) \\
\text{distributivity} & (x \rhd y) - z & = & (x - z) \rhd (y - z) & (5)
\end{array}$$

Figure 1: Axioms for $\rhd$, $-$ and $\emptyset$.

We write $\vdash t = u$ if the equality of t and u can be derived using the standard inference rules of equational logic [4] and the equations in E. We often use the following derived operators:

$$\begin{array}{rrcl}
\text{intersection} & x \mathbin{@} y & = & x - (x - y) \\
\text{update} & x[y] & = & (y \mathbin{@} x) \rhd x
\end{array}$$

Intuitively, operator @ restricts function x to the intersection of the domains of x and y. It allows us to restate Axiom 4 as $x -(y - z) = (x - y) \rhd (x \mathbin{@} z)$. An interpretation of $\cdot[\cdot]$ was already given in the introduction. To ease notations we use the following rules: (a) all operators are left associative, and (b) @ binds strongest, then $-$, and finally $\rhd$. With these conventions, Axiom 4 reads $x -(y - z) = x - y \rhd x \mathbin{@} z$.

Let $\mathcal{F}$ denote any Σ-algebra with as universe some set F of functions, $\rhd$ interpreted as override, $-$ as minus, and $\emptyset$ as the function with the empty domain. We assume F to be closed under these operations.

It is easy to see that $\mathcal{F}$ is a model for the equations of E. The definition of $\rhd$ directly implies that $f \rhd f = f$, for any function f, that is, Axiom 1. By definition of $-$, $f - f$ denotes the restriction of f to the empty domain. This implies Axiom 2. In general, the override operator is not commutative. For instance if f maps a to 1, and g maps a to 0, then $f \rhd g$ maps a to 1, whereas $g \rhd f$ maps a to 0. Axiom 3 states a weak commutativity property that does hold: we may swap the arguments in $x \rhd y$ if we restrict y to the part of its domain that does not intersect with x. Axioms 4 and 5 can be illustrated using the Venn diagrams in Figure 2. Here the colors indicate whether the function value is determined by x, y or z: in the light grey area the value is determined by x and in the dark grey area by y. For each axiom, the left and right hand side of the equation describe different ways in which the same diagram can be obtained.

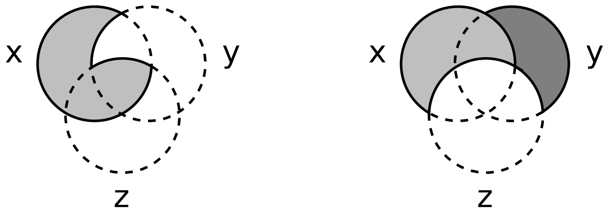

Figure 2: Venn diagrams for Axioms 4 and 5.

Thus we have the following soundness result:

Lemma 1 (Soundness). *Let t, t' be terms over signature Σ such that $\vdash t = t'$. Then Σ-algebra $\mathcal{F}$ satisfies $t = t'$.*

3. Derived Laws

From the basic set of axioms in Figure 1, we derived the laws shown in Figure 3. These laws are used in our proof of completeness. The proofs of all the derived laws have been checked using the proof assistant Isabelle [11]. The Isabelle sources as well as human readable proofs are available as online appendices.

We found it quite surprising that associativity of $\rhd$ (Law 11) and weak commutativity of $-$ (Law 19) are derivable. In the following, we show the proof of associativity of $\rhd$. We use Axioms 2, 3 and 5, and Laws 8, 18, 20, 25, 26, 27, and 28.

Law 11. $x \rhd y \rhd z = x \rhd (y \rhd z)$

Proof. $x \rhd y \rhd z \overset{(25)}{=} (x \rhd y \rhd z) - x \rhd (x \rhd y \rhd z) - ((x \rhd y \rhd z) - x) \overset{(18)}{=} y - x \rhd z - x \rhd (x \rhd y \rhd z) - ((x \rhd y \rhd z) - x) \overset{(5)}{=} (y \rhd z) - x \rhd (x \rhd y \rhd z) - ((x \rhd y \rhd z) - x) \overset{(18)}{=} (y \rhd z) - x \rhd (x \rhd y \rhd z) - (y - x \rhd z - x) \overset{(26)}{=} (y \rhd z) - x \rhd (x \rhd y \rhd z) - (y - x) - (z - x) \overset{(5)}{=} (y \rhd z) - x \rhd ((x \rhd y) - (y - x) \rhd z - (y - x)) - (z - x) \overset{(5)}{=} (y \rhd z) - x \rhd (x - (y - x) \rhd y - (y - x) \rhd z - (y - x)) - (z - x) \overset{(27)}{=} (y \rhd z) - x \rhd (x \rhd y - (y - x) \rhd z - (y - x)) - (z - x) \overset{(28)}{=} (y \rhd z) - x \rhd (x \rhd z - (y - x)) - (z - x) \overset{(27)}{=} (y \rhd z) - x \rhd (x - (y - x) \rhd z - (y - x)) - (z - x) \overset{(5)}{=} (y \rhd z) - x \rhd (x \rhd z) - (y - x) - (z - x) \overset{(19)}{=} (y \rhd z) - x \rhd (x \rhd z) - (z - x) - (y - x) \overset{(5)}{=} (y \rhd z) - x \rhd (x - (z - x) \rhd z - (z - x)) - (y - x) \overset{(27)}{=} (y \rhd z) - x \rhd (x \rhd z - (z - x)) - (y - x) \overset{(28)}{=} (y \rhd z) - x \rhd x - (y - x) \overset{(27)}{=} (y \rhd z) - x \rhd x \overset{(3)}{=} x \rhd (y \rhd z)$.

empty domain	$x @ y - y = \emptyset$	(6)
	$x \triangleright \emptyset = x$	(7)
	$\emptyset \triangleright x = x$	(8)
	$x - \emptyset = x$	(9)
	$\emptyset - x = \emptyset$	(10)
associativity	$x \triangleright (y \triangleright z) = x \triangleright y \triangleright z$	(11)
	$x @ (y @ z) = x @ y @ z$	(12)
idempotence	$x @ x = x$	(13)
distributivity	$(x \triangleright y) @ z = x @ z \triangleright y @ z$	(14)
	$x @ (y \triangleright z) = x @ y \triangleright x @ z$	(15)
	$x @ y - z = (x - z) @ (y - z)$	(16)
	$x \triangleright y @ z = (x \triangleright y) @ (x \triangleright z)$	(17)
	$(x \triangleright y \triangleright z) - x = y - x \triangleright z - x$	(18)
weak commutativity	$x - y - z = x - z - y$	(19)
	$x - (y \triangleright z) = x - (z \triangleright y)$	(20)
	$x - y \triangleright x - z = x - z \triangleright x - y$	(21)
	$x @ y @ z = x @ z @ y$	(22)
	$x - y @ z = x - z @ y$	(23)
partitioning	$x @ y \triangleright x - y = x$	(24)
	$x - y \triangleright x @ y = x$	(25)
combine minus	$(x - y) - z = x - (y \triangleright z)$	(26)
double minus	$x - (y - x) = x$	(27)
overlapping	$x \triangleright y @ x = x$	(28)
agree	$x \triangleright x - y = x$	(29)
	$x - y \triangleright x = x$	(30)
compatible	$x \triangleright y - x = x \triangleright y$	(31)
reordering	$(x - y) @ z = x @ z - y$	(32)
	$x @ (y - z) = x @ y - z$	(33)

Figure 3: Derived laws.

4. Completeness

We now establish completeness of the axioms in set E, that is, we show that if an equality between terms holds in algebra $\mathcal{F}$ then we can actually prove it using the axioms in Figure 1, provided the algebra contains at least all functions from finite subsets of $\mathbb{N}$ to $\mathbb{N}$. We prove completeness by establishing two results: (a) Each term can be reduced to a normal form through application of the axioms. (b) If two terms have different normal forms (modulo associativity and weak commutativity) then they are not equivalent in the algebra, that is, there exists a valuation that maps the two terms to different elements in the domain of the algebra.

Normal forms consist of a combination (with $\triangleright$) of a number of *regions*. A region denotes a function that is defined on the smallest (nonempty) domains that can be defined in our language. Regions are much easier to manipulate than general functions. For instance, if r and r' are regions then $r \triangleright r'$ either equals r (when they have the same domain) or equals $r' \triangleright r$, that is, regions with different domains commute. Below, we formally introduce the concept of a region.

For X a nonempty, finite set of variables, let $C(X)$ be the set of terms defined by the BNF grammar

$$c := x \mid c \mathbin{@} x$$

where $x \in X$, and let $D(X)$ be the set of terms defined by the BNF grammar

$$d := \emptyset \mid d \triangleright x$$

where $x \in X$. A *region* over X is a term of the form $r \equiv c - d$ in which each variable in X occurs exactly once. We say that variable x occurs *positively* in r if it occurs in c, and *negatively* if it occurs in d. We write $pos(r)$ for the set of variables that occur positively in r. Note that at least one variable occurs positively in any region. The first variable occurring in a region is called the *head* or *leading* variable. We write $head(r)$ to denote the head variable of r. We call two regions r and r' over X *equivalent* if they have the same positive variables and the same head, that is, $pos(r) = pos(r')$ and $head(r) = head(r')$. If two regions are equivalent then we can prove their equality using the axioms:

Lemma 2. *Let r and r' be equivalent regions over X. Then $\vdash r = r'$.*

Proof. Let $r \equiv c - d$ and $r' \equiv c' - d'$. Since r and r' are equivalent regions they have the same positive, negative and head variables, and each variable in X occurs only once in r and r'. Hence, the only possible difference between r and r' is the order in which variables occur in d and d', and the order in which non-head variables occur in c and c'. Thus, for proving the lemma, it suffices to show that through application of the axioms we may swap adjacent variables in d and adjacent non-head variables in c. By repeated swapping we can then prove equality of r and r'. Suppose

$$c \equiv x_1 \mathbin{@} \cdots \mathbin{@} x_j \mathbin{@} x_{j+1} \mathbin{@} \cdots \mathbin{@} x_l.$$

Then we can swap non-head variables x_j and x_{j+1} through application of the weak commutativity law (22). Now suppose

$$r \equiv c - (\emptyset \triangleright x_1 \triangleright \cdots \triangleright x_j \triangleright x_{j+1} \triangleright \cdots \triangleright x_l)$$

Then we can swap variables x_j and x_{j+1} by first moving all the brackets in d to the right

$$\overset{(11)}{=} c - (\emptyset \triangleright (x_1 \triangleright (\cdots \triangleright (x_j \triangleright (x_{j+1} \triangleright (\cdots \triangleright x_l))) \cdots)))$$

then replace all $\triangleright$'s preceding x_{j+2} by $-$

$$\overset{(26)}{=} (c - \emptyset - x_1 - \cdots - x_j - x_{j+1}) - (x_{j+2} \triangleright \cdots \triangleright x_l)$$

then swap the two selected variables

$$\overset{(19)}{=} (c - \emptyset - x_1 - \cdots - x_{j+1} - x_j) - (x_{j+2} \triangleright \cdots \triangleright x_l)$$

and then bring the other variables back in place again

$$\begin{aligned}
&\overset{(26)}{=} c-(\emptyset \rhd (x_1 \rhd (\cdots \rhd (x_{j+1} \rhd (x_j \rhd (\cdots \rhd x_l))) \cdots)))\\
&\overset{(11)}{=} c-(\emptyset \rhd x_1 \rhd \cdots \rhd x_{j+1} \rhd x_j \rhd \cdots \rhd x_l).
\end{aligned}$$

Lemma 3. *Let r and r' be regions over X.*

1. *If $pos(r) = pos(r')$ then $\vdash r - r' = \emptyset$.*
2. *If $pos(r) \neq pos(r')$ then $\vdash r - r' = r$.*
3. *If $pos(r) = pos(r')$ then $\vdash r \rhd r' = r$.*
4. *If $pos(r) \neq pos(r')$ then $\vdash r \rhd r' = r' \rhd r$.*

Proof.

1. Suppose $pos(r) = pos(r')$. Let $r \equiv c - d$ and $r' \equiv c' - d'$. Then using the same tricks as in the proof of Lemma 2, we can reorder the subterms of d' to obtain $\vdash c' - d' = c' - d$. Also, since c and c' only differ in the ordering of their variables, we can use Laws 12, 22 and 23 to reorder the variables in c' and obtain $\vdash c - c' = c - c$. Hence

$$\begin{aligned}
r - r' &\equiv (c-d) - (c'-d')\\
&= (c-d) - (c'-d)\\
&\overset{(26)}{=} c - (d \rhd (c'-d)) \overset{(31)}{=} c - (d \rhd c')\\
&\overset{(20)}{=} c - (c' \rhd d) \overset{(26)}{=} (c - c') - d\\
&= (c-c) - d \overset{(2)}{=} \emptyset - d \overset{(10)}{=} \emptyset.
\end{aligned}$$

2. Suppose $pos(r) \neq pos(r')$. Then there exists a variable x that occurs positively in r and negatively in r', or vice versa. If x occurs positively in a region u then $\vdash u = u @ x$. In order to see this, let $u \equiv c - d$. Using Laws 13, 12 and 22, we can prove $c = c @ x$. By Law 32 we derive, $u = c @ x - d = (c-d) @ x = u @ x$. If x occurs negatively in a region u then $\vdash u = u - x$. In order to see this, let $u \equiv c - d$. Using Axiom 1 and the tricks from Lemma 2, we can prove that $u = c - (d \rhd x)$. Hence, by Law 26, $u = (c-d) - x = u - x$. Thus, if x occurs positively in r we may infer

$$r - r' = r @ x - (r' - x) \overset{(33)}{=} r @ (x - (r' - x)) \overset{(27)}{=} r @ x = r.$$

If x occurs negatively in r then we may infer

$$r - r' = (r - x) - (r' @ x) \overset{(26)}{=} r - (x \rhd r' @ x) \overset{(28)}{=} r - x = r.$$

3. Suppose $pos(r) = pos(r')$. Then

$$r \rhd r' \overset{(3)}{=} (r' - r) \rhd r \overset{\text{Lemma 3(1)}}{=} \emptyset \rhd r \overset{(8)}{=} r.$$

4. Suppose $pos(r) \neq pos(r')$. Then

$$r \rhd r' \overset{(3)}{=} (r' - r) \rhd r \overset{\text{Lemma 3(2)}}{=} r' \rhd r.$$

We say that a term is a *normal form* over X if it is of the form

$$r_1 \rhd \cdots \rhd r_m$$

where, for all $1 \leq i \leq m$, r_i is a region over X and, for all $1 \leq i < j \leq m$, $pos(r_i) \neq pos(r_j)$. By convention, $\emptyset$ is a normal form ($m = 0$) and a single region is a normal form ($m = 1$). We call two normal forms n and n' over X *equivalent* if for each region contained in n there is an equivalent region contained in n', and vice versa.

Lemma 4. *Let n and n' be equivalent normal forms over X. Then $\vdash n = n'$.*

Proof. By Lemma 2, we know that whenever two regions r and r' are equivalent we can prove that they are equal, that is $\vdash r = r'$. Thus there exists a normal form n'' such that $\vdash n' = n''$ and each region contained in n'' is also a region of n, and vice versa. This means that n and n'' only differ in the ordering of their regions. But by associativity and Lemma 3(4), the ordering of regions in a normal form does not matter, that is, $\vdash n = n''$. Hence $\vdash n = n'$, as required.

The following technical lemma is needed to prove that any term can be reduced to a normal form.

Lemma 5. *Let n be a normal form over X and let x be a variable with $x \notin X$. Then there exists a normal form n' over $X \cup \{x\}$ such that $\vdash n = n'$.*

Proof. Suppose $r \equiv c - d$ is a region over X. Then $r \mathbin{@} x$ and $r - x$ are provably equal to regions over $X \cup \{x\}$:

$$\begin{aligned} r \mathbin{@} x \equiv (c - d) \mathbin{@} x &\overset{(32)}{=} c \mathbin{@} x - d \\ r - x \equiv (c - d) - x &\overset{(26)}{=} c - (d \rhd x). \end{aligned}$$

Let $n \equiv r_1 \rhd \cdots \rhd r_m$. Then

$$\begin{aligned} n &\equiv r_1 \rhd \cdots \rhd r_m \\ &\overset{(24)}{=} (r_1 \rhd \cdots \rhd r_m) \mathbin{@} x \ \rhd\ (r_1 \rhd \cdots \rhd r_m) - x \\ &\overset{(14),(5)}{=} (r_1 \mathbin{@} x \rhd \cdots \rhd r_m \mathbin{@} x) \ \rhd\ (r_1 - x \rhd \cdots \rhd r_m - x) \end{aligned}$$

By the above observation, all subterms $r_i \mathbin{@} x$ and $r_i - x$ can be replaced by regions over $X \cup \{x\}$. Let n' denote the resulting term. Then n' is a normal form over $X \cup \{x\}$ such that $\vdash n = n'$.

The next lemma states that any term can be reduced to a normal form.

Lemma 6. *Let t be a term over signature Σ with variables contained in X. Then there exists a normal form n over X such that $\vdash t = n$.*

Proof. By induction on the structure of term t:

1. If $t \equiv \emptyset$ then t is a normal form already and, by reflexivity, we have $\vdash t = t$.
2. If $t \equiv x$ then, by Law 9, $\vdash t = x - \emptyset$. Term $x - \emptyset$ is a normal form over $\{x\}$. By repeated application of Lemma 5, we can rewrite $x - \emptyset$ into a normal form over X, as required.
3. If $t \equiv t_1 \rhd t_2$ then, by induction hypothesis, there exist normal forms n_1 and n_2 such that $\vdash t_1 = n_1$ and $\vdash t_2 = n_2$. Using associativity of $\rhd$, Lemma 3(3) and Lemma 3(4), we can eliminate all occurrences of regions in $n_1 \rhd n_2$ that are preceded with a region that has the same positive variables. The resulting term n is a normal form n over X such that $\vdash t = n$.
4. If $t \equiv t_1 - t_2$ then, by induction hypothesis, there exist normal forms n_1 and n_2 such that $\vdash t_1 = n_1$ and $\vdash t_2 = n_2$. Let $n_1 \equiv r_1 \rhd \cdots \rhd r_k$ and $n_2 \equiv r'_1 \rhd \cdots \rhd r'_l$. Then

$$\begin{aligned} n_1 - n_2 &\equiv (r_1 \rhd \cdots \rhd r_k) - n_2 \\ &\overset{(5)}{=} r_1 - n_2 \rhd \cdots \rhd r_k - n_2 \\ &\overset{(26)}{=} r_1 - r'_1 - r'_2 - \cdots - r'_l \rhd \cdots \rhd r_k - r'_1 - r'_2 - \cdots - r'_l. \end{aligned}$$

   Using Lemma 3(1), Lemma 3(2), and the laws for $\emptyset$, we may reduce each subterm in the above expression to either a region of n_1 or to $\emptyset$. After elimination of spurious $\emptyset$'s, the resulting term n is a normal form over X such that $\vdash t = n$.

The following lemma states that if two normal forms are not equivalent, they are in fact not equal in any (nontrivial) algebra.

Lemma 7. *Let n, n' be normal forms over X that are not equivalent. Let $\mathcal{F}$ be a Σ-algebra with as universe a set of functions that contains at least all functions from finite subsets of $\mathbb{N}$ to $\mathbb{N}$. Then $\mathcal{F}$ does not satisfy equation $n = n'$.*

Proof. Let $X = \{x_1, \ldots, x_k\}$. Assume w.l.o.g. that n contains a region r which is, up to equivalence, not contained in n'. We define functions $f_1, \ldots, f_k$ such that any valuation ξ that assigns f_i to x_i, for $1 \leq i \leq k$, maps n to a different element of $\mathcal{F}$ than n'. Let $D = \{0, 1, \ldots, 2^k - 1\}$. Let $\mathsf{binary} : D \to \mathbb{B}^k$ be the function that assigns to each number in D its binary encoding. Define

$$\mathsf{dom}(f_i) \triangleq \{j \in D \mid i\text{-th bit in } \mathsf{binary}(j) \text{ equals } 1\}$$

and let $f_i(v) \triangleq i$, for all $v \in \mathsf{dom}(f_i)$. Let ξ be a valuation that maps x_i to f_i, for $1 \leq i \leq k$. Then each region corresponds to a function whose domain is a singleton set and which maps the unique element in its domain to the index of the head variable of the region. Moreover, the domain of each region is uniquely determined by its positive variables. Since n contains a region r that, up to equivalence, is not contained in n':

- either n' contains a region r' with $pos(r) = pos(r')$ and $head(r) \neq head(r')$,
- or n' does not contain a region r' with $pos(r) = pos(r')$.

In the first case $\mathcal{F}$ does not satisfy $n = n'$ since valuation ξ maps the terms to functions that differ for at least one element in their domain. In the second case $\mathcal{F}$ does not satisfy $n = n'$ since ξ maps the two normal forms to functions with different domains.

Theorem 8 (Completeness). *Let $\mathcal{F}$ be a Σ-algebra with as universe a set of functions that contains at least all functions from finite subsets of $\mathbb{N}$ to $\mathbb{N}$. Let t, t' be terms over signature Σ such that algebra $\mathcal{F}$ satisfies equation $t = t'$. Then $\vdash t = t'$.*

Proof. Let X be a finite set of variables that contains the variables in t and t'. By Lemma 6, there exist normal forms n and n' over X such that $\vdash t = n$ and $\vdash t' = n'$. By Lemma 1, $\mathcal{F}$ satisfies equations $t = n$ and $t' = n'$. Using the fact that $\mathcal{F}$ satisfies $t = t'$, we infer that $\mathcal{F}$ satisfies $n = n'$. Now Lemma 7 implies that normal forms n and n' are equivalent. Thus, by Lemma 4, $\vdash n = n'$. Combining this with $\vdash t = n$ and $\vdash t' = n'$ yields $\vdash t = t'$, as required.

5. Independence

In this section, we are going to prove that the axioms in Figure 1 are independent. One common way to prove independence of an axiom is to come up with an algebra that violates this axiom but satisfies the other ones. Using this technique it is not difficult to prove independence of the first three axioms.

Proposition 9. *Axiom 1 is independent.*

Proof. Consider a model with a domain consisting of two elements $D \triangleq \{\top, \bot\}$, in which all operators always return the first element, that is $\forall v, w \in D : v \rhd w = v - w = \emptyset = \top$. Then all axioms hold trivially, except for the idempotence axiom $x \rhd x = x$, which does not hold in case x is assigned the value $\bot$.

Proposition 10. *Axiom 2 is independent.*

Proof. Consider a model with a domain consisting of two elements $D \triangleq \{\top, \bot\}$, in which $\rhd$ is interpreted as logical or, $v - w = v$ for all $v, w \in D$, and $\emptyset = \bot$. Then all axioms hold trivially, except for the axiom $x - x = \emptyset$, which does not hold in case x is assigned the value $\top$.

Proposition 11. *Axiom 3 is independent.*

Proof. Consider again a model with domain $D \triangleq \{\top, \bot\}$. This time operator $\rhd$ always returns its first operand, i.e. $v \rhd w = v$ for all $v, w \in D$, operator $-$ always returns the top element, i.e. $v - w = \top$ for all $v, w \in D$, and $\emptyset = \top$. Then all axioms hold trivially, except for the axiom $x \rhd y = (y - x) \rhd x$, which does not hold in case x has value $\bot$.

In order to prove independence of Axioms 4 and 5, we used the SMT solver Yices [17]. For this problem we could also have used a regular SAT solver, but we found the input language of Yices more convenient. Peter Jipsen discovered exactly the same models using Prover9/Mace4 [13]. In our encoding, we assumed that the model that establishes independence of an axiom has a finite domain. Each axiom was encoded as a proposition on this domain. We asked the solver whether there exist any possibilities for $\triangleright$ and $-$ that show all axioms to hold, except for the one we were trying to prove independent. Note that for a domain of n elements an operator already has $n^{n \cdot n}$ possibilities, because it has $n \cdot n$ possible input combinations, that all lead to an element of the domain.

Proposition 12. *Axiom 4 is independent.*

$\triangleright$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

$-$	0	1	2
0	0	0	0
1	1	0	0
2	2	2	0

Figure 4: Model that establishes independence of Axiom 4.

Proof. Consider a model with a domain consisting of three elements $D \triangleq \{0, 1, 2\}$. Let operators $\triangleright$ and $-$ work as specified by the tables in Figure 4, and let $\emptyset = 0$. Note that operator $\triangleright$ returns the maximum of its arguments. This model was found using Yices. All axioms hold in this model, except Axiom 4, which does not hold if x and z have value 1, and y has value 2:

$$1 - (2 - 1) = 1 - 2 = 0 \neq 1 = 0 \triangleright 1 = 0 \triangleright 1 - 0 = 1 - 2 \triangleright 1 - (1 - 1).$$

An isomorphic model can be obtained by slightly changing the standard, functional model $\mathcal{F}$ introduced in Section 2. We assume a setting where the set F of functions contains three elements: the function 0 which has the empty domain, a function 2 with a domain of at least two elements, and a function 1 that is the restriction of 2 to a strict subdomain with at least one element. As in the standard model, operator $\triangleright$ is interpreted as override and $\emptyset$ as 0. Operator $-$ is interpreted as minus, but since the difference $2 - 1$ cannot be represented exactly it is "approximated" by 2.

Proposition 13. *Axiom 5 is independent.*

$\triangleright$	0	1	2
0	0	1	2
1	1	1	0
2	2	0	2

$-$	0	1	2
0	0	0	0
1	1	0	1
2	2	2	0

Figure 5: Model that establishes independence of Axiom 5.

Proof. Consider the model in Figure 5, with 0 as the empty function. Also this model was found using Yices. In this model,

$$(1 \triangleright 2) - 1 = 0 - 2 = 0 \neq 2 = 0 \triangleright 2 = 1 - 1 \triangleright 2 - 1.$$

Again, an isomorphic model can be obtained by slightly changing the standard, functional model $\mathcal{F}$. The set F of functions contains three elements: the function 0 which has the empty domain, and functions 1 and 2 that have nonempty, disjoint domains. As in the standard model, operator $-$ is interpreted as minus and $\emptyset$ as 0. Operator $\triangleright$ is interpreted as override, but since the union ($1 \triangleright 2 = 2 \triangleright 1$) cannot be represented exactly it is "approximated" by the "neutral" element 0.

6. Functional Completeness

The reader may ask why we have chosen the operators override and minus to describe the operations that are possible on general functions. In this section, we want to show that override and minus are *functionally complete* in the sense that any operation on general functions that corresponds to a valid coloring of a Venn diagram can be described using just these two operations. Our result is a simple extension of the well-known functional completeness results for Boolean operations.

One can visualize a Boolean operation on the (Boolean) variables $x_1, \dots, x_n$ by a shading of a Venn diagram: Draw one circle (or shape) for every variable. The interior and the exterior of the circle for x_i correspond to x_i being true and false, respectively. The shading of areas in the diagram indicates the result of the Boolean operation. All n-ary operations $f : \mathbb{B}^n \to \mathbb{B}$ can be reduced to a small number of binary operations (see e. g., [7, Theorem 7.12]).

Similarly, one can visualize an operation on functions as a multicolored Venn diagram: we start with one shape for each function variable $x_1, \dots, x_n$, and each of the areas may be colored with one of the variables defined in that area. Figure 2 shows two multicolored Venn diagrams. We can reduce all n-ary operations on functions that result from coloring Venn diagrams to override and minus.

We now formalize the notions introduced above. Assume a set of n function variables $V = \{x_1, \dots, x_n\}$. A *valid coloring* of the n-ary Venn diagram is a partial mapping c from the subsets of V to V, where each $W \subseteq V$ in the domain of c is mapped to some element of W.

The valid coloring c corresponds to the following operation on functions: $(f_1, \dots, f_n)$ is mapped to f_c with the property: $f_c(p) = f_i(p)$ if there is some $W \subseteq V$ such that $c(W) = x_i$ and p lies in the intersection of domains described by W, that is, $p \in \mathsf{dom}(f_j)$ iff $x_j \in W$, for all j. The update operator $x_1[x_2]$, for example, can be described by the coloring c that satisfies

$$\begin{aligned} \{x_1\} &\mapsto x_1 \\ \{x_1, x_2\} &\mapsto x_2 \end{aligned}$$

and that is undefined for $\emptyset$ and $\{x_2\}$.

The next proposition states that override and minus are functionally complete.

Proposition 14. *Every n-ary operation on functions that corresponds to some valid coloring c (for $n \geq 2$) can be described by a term composed of variables $\{x_1, \dots, x_n\}$ and binary function symbols $\rhd$ and $-$.*

Proof. Assume given the valid coloring c, a partial mapping from subsets of $V = \{x_1, \dots, x_n\}$ to V. The statement is easily proven using the normal form: If $c(W)$ is defined, assign to W a region over V whose head is $c(W)$, whose positive variables are the ones in W, and whose negative variables are the ones in $V \setminus W$. Then, the normal form that is the composition of all such terms (with $\rhd$) describes the desired operation on functions. Finally, we can eliminate all occurrences of $\emptyset$ in the normal form by replacing them by the term $x_1 - x_1$.

Observe that override and update are not functionally complete. Consider any n-ary function f that is defined using override and update. Then the domain of f is the union of the domains of some of its arguments. Thus, for instance, it is not possible to define the minus operator in terms of override and update.

One important difference between functional completeness of Boolean connectives and of function connectives is the following: All function connectives are *false-preserving*, i.e., they necessarily map $(\emptyset, \dots, \emptyset)$ to $\emptyset$. Stated in terms of Venn diagrams, the outermost area must remain uncolored. According to Post's classification of Boolean operators [12], this implies that Boolean operators corresponding to $\rhd$ and $-$ (or whichever set of function operators one chooses) are *not* functionally complete for Boolean operations.

7. Conclusions and Future Work

In this paper, we have presented a finite, equational axiomatization of override and minus, two fundamental operators for combining arbitrary functions. We have established soundness and completeness of our axiomatization, proved that the set of equations is minimal, and established functional completeness of override and minus.

The top level structure of the completeness proof is not very surprising: reduction to normal form and a proof that distinct normal forms have a distinct semantics. The surprising aspect of our result is that it can be proven with only five laws. It turns out that the algebra induced by these laws is extremely rich. Many long derivations of auxiliary laws and identities are needed for the normal form theorem, see for instance the derivation of Law 11 in Section 3 and the proofs of the other laws in Figure 3 (see the online appendices for the actual proofs.). Finding and proving these auxiliary laws and identities is what made the completeness proof difficult.

An interesting aspect of our work is that two computerized tools, the proof assistant Isabelle and the SMT solver Yices, were essential for us to obtain our results. Isabelle found the nontrivial proof of Law 26, which plays an important role in the completeness proof. Yices established the independence of Axioms 4 and 5. This nicely illustrates the growing importance of computerized tools for proving mathematical theorems. A challenging case study would be to formalize our proofs of soundness, completeness and minimality in Isabelle.

Our initial research question was to come up with a complete axiomatization of override and update. We have solved this problem by adding the auxiliary minus operator. A natural question that remains open is whether it is possible to axiomatize override and update directly without auxiliary operators. Candidate laws include associativity and idempotence of $\rhd$, and

$$x[x \rhd y] = x \tag{34}$$

$$x \rhd y = y[x] \rhd x \tag{35}$$

$$x[y][z] = x[z \rhd y] \tag{36}$$

$$(x \rhd y)[z] = x[z] \rhd y[z] \tag{37}$$

Are these laws complete? If not, which laws should be added? Does there exist a finite equational axiomatization of override and update without auxiliary operators? All these questions are open to us. Other open questions are how to build a term rewriting system out of our axioms, and the generation of normal forms with minimal size.

Our completeness proof is proof theoretic (syntactic). It would be very interesting to explore the model theory of our logic. We have, for instance, shown that using our axioms one can derive associativity of override, but we still do not have a deep understanding why this is the case. We expect that study of the model theory of override and update, that is, study of the interplay of syntactic and semantic ideas, may provide more insight [5]. Another interesting research direction would be to study the override and update operators within the setting of category theory. In the category **Set** disjoint union can be presented as a pushout and the override operator can be expressed naturally in terms of disjoint union and minus: $x \rhd y = x + (y - x)$.

As an interesting alternative model for our axioms, one may consider the Boolean algebra of sets, but with complement replaced by relative complement (set difference). The domain of this model consists of all subsets of a given set S, $\rhd$ is interpreted as set union, $-$ as set difference, and $\emptyset$ as the empty set. The derived operator @ corresponds to intersection and update becomes trivial: $x[y] = x$. Whereas the Boolean algebra of sets and its equational axiomatization is standard textbook material, the axiomatization of its relativized version is less known. As pointed out by Mai Gehrke, an axiomatization can be obtained from the work of Stone, and Balbes and Dwinger [2, 15]. Balbes and Dwinger [2][Def.3 in II.7, p. 55] define a *relatively complemented distributive lattice* to be a distributive lattice in which every element is relatively complemented, and a *generalized Boolean algebra* to be a relatively complemented distributive lattice with a 0. They show that those are exactly the lattices corresponding to Boolean rings (where, as usual, the sum is symmetric difference and the product is the meet). This result provides an equational axiomatization. Balbes and Dwinger also give an axiomatization for relatively complemented distributed lattices using a ternary operation. An alternative completeness proof for generalized Boolean algebra was found recently by Clemens Grabmayer and Albert Visser [9].

Laws for override and update played a key role in our work on compositional abstraction for timed automata [3], in particular in the proof of associativity of parallel composition, thus illustrating the practical usefulness of our work. We expect that our axiomatization will also be useful in the settings of Z [14] and VDM [10].

Acknowledgment. We are grateful to Jean-François Raskin for encouraging us to prove completeness of our axiomatization. We thank Mai Gehrke and Peter Jipsen for insightful comments, and Christoph Sprenger and an anonymous referee for careful proofreading.

[1] J. de Bakker, Mathematical Theory of Program Correctness, Prentice-Hall, Englewood Cliffs, 1980.

[2] R. Balbes, P. Dwinger, Distributive Lattices, University of Missouri Press, Columbia, 1974.

[3] J. Berendsen, F. Vaandrager, Compositional abstraction in real-time model checking, Tech. Rep. ICIS–R07027, Radboud University, Nijmegen, an extended abstract appeared in *Proc. FORMATS'08*, LNCS 5215 (2007).

[4] G. Birkhoff, On the structure of abstract algebras, Proceedings of the Cambridge Philosophical Society 31 (4) (1935) 433–454.

[5] C. C. Chang, H. J. Keisler, Model Theory, 3rd Edition, Vol. 73 of Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1990.

[6] W. Damm, B. Josko, H. Hungar, A. Pnueli, A compositional real-time semantics of STATEMATE designs, in: W.-P. de Roever, H. Langmaack, A. Pnueli (Eds.), Compositionality, the significant difference: . . . COMPOS '97, Vol. 1536 of Lecture Notes in Computer Science, Springer, Berlin, 1998, pp. 186–238.

[7] B. A. Davey, H. A. Priestley, Introduction to Lattices and Order, Cambridge University Press, Cambridge, UK, 1990.

[8] A. V. Goldberg, R. E. Tarjan, A new approach to the maximum-flow problem, Journal of the ACM 35 (4) (1988) 921–940.

[9] C. Grabmeyer, A. Visser, Axiomatization of generalized boolean algebras, personal communication (2007).

[10] C. Jones, Systematic Software Development using VDM, Prentice-Hall, Englewood Cliffs, 1986.

[11] T. Nipkow, L. C. Paulson, M. Wenzel, Isabelle/HOL: A Proof Assistant for Higher-Order Logic, Vol. 2283 of Lecture Notes in Computer Science, Springer, Berlin, 2002.

[12] E. L. Post, The two-valued iterative systems of mathematical logic, no. 5 in Annals of mathematics studies, Princeton University Press, Princeton, 1941.

[13] Prover9/Mace4, `http://www.cs.unm.edu/~mccune/prover9/`.

[14] J. M. Spivey, The Z notation: a reference manual, Prentice-Hall, Englewood Cliffs, 1989.

[15] M. H. Stone, Subsumption of boolean algebras under the theory of rings, Proceedings of the National Academy of Sciences of the USA 21 (2) (1935) 103–105.

[16] G. Winskel, The Formal Semantics of Programming Languages: An Introduction, MIT Press, Cambridge, Mass., 1993.

[17] Yices, `http://yices.csl.sri.com`.