Axiomatic Set Theory à la Dijkstra and Scholten

Ernesto Acosta[1], Bernarda Aldana[1], Jaime Bohórquez[1], and Camilo Rocha[2]*

[1] Escuela Colombiana de Ingeniería Julio Garavito, Bogotá, Colombia
[2] Pontificia Universidad Javeriana, Cali, Colombia

Abstract. The algebraic approach by E. W. Dijkstra and C. S. Scholten to formal logic is a proof calculus, where the notion of proof is a sequence of equivalences proved – mainly – by using substitution of 'equals for equals'. This paper presents Set, a first-order logic axiomatization for set theory using the approach of Dijkstra and Scholten. The approach is novel in that the symbolic manipulation of formulas is shown to be an effective tool for teaching axiomatic set theory to sophomore students in mathematics. This paper contains many examples on how argumentative proofs can be easily expressed in Set and points out how Set can enrich the learning experience of students. These results are part of a larger effort to formally study and mechanize topics in mathematics and computer science with the algebraic approach of Dijkstra and Scholten.

Keywords: Axiomatic set theory · Dijkstra-Scholten logic · Derivation · Formal system · Zermelo-Fraenkel (ZF) · Symbolic manipulation · Undergraduate-level course.

1 Introduction

Axiomatic set theory is the branch of mathematics that studies collections of objects from the viewpoint of mathematical logic. In general, axiomatic set theory focuses on the properties of the membership relation '$\in$', given the existence of some basic sets (e.g., the empty set). Unlike 'naive' set theory – where definitions are given in natural language, and Venn diagrams and Boolean algebra are used to support reasoning about collections – the axiomatic study of sets begins with a set of axioms and then associates axiomatic rules to suitably defined sets and constructive relations. Because other theories across different branches of mathematics (e.g., number theory, topology) can be encoded in set theory, it plays an important role as foundational system.

An axiomatic theory for sets is usually given as a first-order logic theory, i.e., as a formal system that uses: (i) universally and existentially quantified variables over non-logical objects, and (ii) formulas that can contain variables, function symbols, and predicate symbols. Variables range over collections; function symbols include the empty set, projections, and cardinality; and predicate symbols include membership and equality. The Zermelo-Fraenkel (ZF) is the most common axiomatic set theory [7], sometimes including the axiom of choice (ZFC), which aims at formalizing the notion of *pure set*

* The first three authors have been supported in part by grant DII/C004/2015 funded by Escuela Colombiana de Ingeniería.

or *hereditary well-founded set* so that all entities in the universe of discourse are such collections.

This paper presents an axiomatization for set theory using the *calculational* approach developed by E. W. Dijkstra and C. S. Scholten to formal logic [1]. Its main contribution is a first-order theory for sets having as its key feature the symbolic manipulation of formulas under the principle known as Leibniz's rule: the substitution of 'equals for equals'. While there are many deductive systems for first-order logic, both sound (i.e., all provable statements are true in all models) and complete (i.e., all statements which are true in all models are provable), the notion of proof in the Dijkstra-Scholten logic focuses on logical equivalence rather than implication. In general, the Dijkstra-Scholten logic can be seen as a correct choice of connectives, axioms, and inference rules, allowing for proofs of logical formulas by symbol manipulation, without the need for introducing unnecessary assumptions.

The algebraic approach by E. W. Dijkstra and C. S. Scholten to formal logic, in general, is a *proof calculus* [8]. For the proposed axiomatic theory, Dijkstra-Scholten refers to the logical system resulting from the combination of first-order logic and their proof style. The notion of 'proof' in the Dijkstra-Scholten system is actually a *derivation* [10], i.e., a sequence of equivalences proved, mainly, by using Leibniz principle: if two formulas are provably equivalent, then substituting one for the other does not alter the meaning of any formula. In a nutshell, a derivation can follow various approaches and can always be translated into a formal proof [10] (e.g., in a Hilbert-like system). For an axiomatic theory of sets, derivations result in rigorous and elegant counterparts to argumentative proofs, commonly found in textbooks, which can help proving theorems succinctly.

The results presented in this paper are a partial report on a two-term seminar experience to solve all exercises and symbolically rewrite all proofs in sections 1, 2, and 9 of Chapter 1 in [6], using the Dijkstra-Scholten approach. The meta-mathematical aspects of set theory such as the aspects of semantics, completeness, and axiom independence are not considered since the main interest is to realize how to teach formal thinking to undergraduates. A sophomore-year course with this approach has been successfully taught twice.

This work is also a part of a larger effort to formally study and mechanize topics in mathematics and computer science with the algebraic approach of E. W. Dijkstra and C. S. Scholten. In particular, the set theory axiomatization of ZF in first-order logic presented in this paper is the first step towards a mechanization in rewriting logic [9], a logic in which concurrent rewriting coincides with logical deduction. What is appealing about mechanizing theories à la Dijkstra-Scholten is that they are written in a relatively strict format that can be easily accessed by humans (which is seldom the case with most tools). In the setting of rewriting logic, the notion of substitution of 'equals for equals' is a natural part of deduction because it is a more general case of equational logic.

To sum up, the main contributions are:

- a set theory axiomatization of ZF in first-order logic using the calculational style of E. W. Dijkstra and C. S. Scholten;
- examples of some proofs obtained by using derivations, compared to their argumentative versions found in textbooks; and

– a discussion on how student experience has improved in a sophomore-year undergraduate-level axiomatic set theory course taught à la Dijkstra and Scholten .

The rest of the paper is organized as follows. Section 2 presents the first-order Dijkstra-Scholten system. Section 3 presents the axiomatic set theory à la Dijkstra-Scholten and Section 4 presents examples of proofs in this theory based on derivations. Section 5 presents a discussion explaining how the use of the proposed approach has helped in teaching an undergraduate-level course on axiomatic set theory. Finally, Section 6 presents some related work and concluding remarks.

2 The Formal System of Dijkstra and Scholten

This section overviews the Dijkstra and Scholten first-order formal system, summarizing sections 2-5 in [10].

A formal system uses an alphabet to construct a formal language from a collection of axioms through inferential rules of formation. More precisely, a formal system [3] consists of: a (possibly infinite) collection of symbols or alphabet; a grammar defining how well-formed formulas are constructed based on symbols in the alphabet; a collection of axioms; and a collection of inference rules. The collections of formulas of the formal systems presented here, all have a decidable membership problem.

Definition 1. *Let* F *be a formal system,* Γ *a collection of* F*-formulas, and* φ_n *a* F*-formula. A* proof of φ_n from Γ in F *is a sequence of* F*-formulas* $\varphi_0, \varphi_1, \ldots, \varphi_n$ *such that for any* $0 \leq i \leq n$*: (i)* φ_i *is an axiom, (ii)* $\varphi_i \in \Gamma$*, or (iii)* φ_i *is the conclusion of an inference rule with premises appearing in* $\varphi_0, \ldots, \varphi_{i-1}$*. An* F*-formula* φ is a theorem from Γ in F*, denoted as* $\Gamma \vdash_{\mathsf{F}} \varphi$*, if and only if there is a proof of* φ *from* Γ *in* F*; in the case when* $\Gamma = \emptyset$*,* φ *is called a* theorem of F*, denoted as* $\vdash_{\mathsf{F}} \varphi$*.*

The first-order system of E. W. Dijkstra and C. S. Scholten (with equality) is presented as the formal system $\mathsf{DS}(\mathcal{L})$, which is parametric on a first-order language $\mathcal{L}$.

Definition 2. *The symbols of* $\mathsf{DS}(\mathcal{L})$ *are:*

– *An infinite collection* $\mathcal{X}$ *of* variables $x_0, x_1, x_2, \ldots$.
– *A collection* $\mathcal{F}$ *of* function symbols.
– *A collection* $\mathcal{P}$ *of* predicate symbols*, which includes infinitely many* constants $P_0, P_1, P_2, \ldots$.
– *An* arity function $ar : \mathcal{F} \cup \mathcal{P} \to \mathbb{N}$ *for function and predicate symbols.*
– *Left parenthesis '(', right parenthesis ')', and comma ','.*
– *The* logical connectives $true, false, \neg, \equiv, \not\equiv, \vee, \wedge, \to, \leftarrow, \forall, \exists$.

The infinitely many constant predicate symbols assumed to be in $\mathcal{P}$ are key for formula manipulation in the formal system. The logical connectives of $\mathsf{DS}(\mathcal{L})$ include the Boolean constants *true* and *false*, negation '$\neg$', equivalence '$\equiv$', discrepancy '$\not\equiv$', disjunction '$\vee$', conjunction '$\wedge$', implication '$\to$', consequence '$\leftarrow$', '$\forall$' for universal quantification, and '$\exists$' for existential quantification.

Terms and formulas in $\mathsf{DS}(\mathcal{L})$ are built in the usual way. A term is built from variables and the application of a function symbol to a sequence of terms. A formula is built from the Boolean constants, term equality, and Boolean combination of formulas, with the application of a predicate symbol to a sequence of terms and universal/existential quantified formulas. The *atomic formulas* of $\mathsf{DS}(\mathcal{L})$ are the Boolean constants *true* and *false*, equality of terms, and the formulas obtained by applying a predicate symbol to zero or more terms. Definition 3 introduces the terms and formulas of $\mathsf{DS}(\mathcal{L})$.

Definition 3. *The collection of* terms *and the collection of* formulas *of the formal system* $\mathsf{DS}(\mathcal{L})$ *are given by the following BNF definitions, where* $x \in \mathcal{X}$, $c \in \mathcal{F}$ *with* $ar(c) = 0$, $f \in \mathcal{F}$ *with* $ar = m > 0$, $P \in \mathcal{P}$ *with* $ar(P) = 0$, $Q \in \mathcal{P}$ *with* $ar(Q) = n > 0$, t *is a term, and* φ *is a formula:*

$$\begin{aligned} t &::= x \mid c \mid f(t, \ldots, t) \\ \varphi &::= true \mid false \mid t = t \mid P \mid Q(t, \ldots, t) \mid (\neg\varphi) \mid (\varphi \equiv \varphi) \mid (\varphi \not\equiv \varphi) \mid (\varphi \vee \varphi) \\ &\quad \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid (\varphi \leftarrow \varphi) \mid (\forall x\, \varphi) \mid (\exists x\, \varphi)\,. \end{aligned}$$

The expressions $\mathcal{T}(\mathcal{X}, \mathcal{F})$ *and* $\mathcal{T}(\mathcal{X}, \mathcal{F}, \mathcal{P})$ *denote the collection of terms and the collection of formulas over* $\mathcal{X}$, $\mathcal{F}$, *and* $\mathcal{P}$, *respectively.*

In the Dijkstra-Scholten first-order logic, the textual substitution operator $\_[\_ := \_]$ is overloaded both for replacing variables for terms and for replacing constant predicate symbols for formulas. The concept of a free occurrence of a variable in a formula in the Dijkstra-Scholten logic is the traditional one, i.e., an occurrence of a variable x in a formula φ is *free* iff such an occurrence of x is not under the scope of a $\forall x$ or $\exists x$. Similarly, a term t is *free* for x in a formula φ iff every free occurrence of x in φ is such that if it is under the scope of a $\forall y$ or $\exists y$, then y is not a variable in t.

Definition 4. *Let* $x \in \mathcal{X}$, $t \in \mathcal{T}(\mathcal{X}, \mathcal{F})$, *and* $\varphi, \psi \in \mathcal{T}(\mathcal{X}, \mathcal{F}, \mathcal{P})$. *The* collection of axioms of $\mathsf{DS}(\mathcal{L})$ *is given by the following axiom schemata:*

$(Ax1)$ $((\varphi \equiv (\psi \equiv \tau)) \equiv ((\varphi \equiv \psi) \equiv \tau))$.
$(Ax2)$ $((\varphi \equiv \psi) \equiv (\psi \equiv \varphi))$.
$(Ax3)$ $((\varphi \equiv true) \equiv \varphi)$.
$(Ax4)$ $((\varphi \vee (\psi \vee \tau)) \equiv ((\varphi \vee \psi) \vee \tau))$.
$(Ax5)$ $((\varphi \vee \psi) \equiv (\psi \vee \varphi))$.
$(Ax6)$ $((\varphi \vee false) \equiv \varphi)$.
$(Ax7)$ $((\varphi \vee \varphi) \equiv \varphi)$.
$(Ax8)$ $((\varphi \vee (\psi \equiv \tau)) \equiv ((\varphi \vee \psi) \equiv (\varphi \vee \tau)))$.
$(Ax9)$ $((\neg\varphi) \equiv (\varphi \equiv false))$.
$(Ax10)$ $((\varphi \not\equiv \psi) \equiv ((\neg\varphi) \equiv \psi))$.
$(Ax11)$ $((\varphi \wedge \psi) \equiv (\varphi \equiv (\psi \equiv (\varphi \vee \psi))))$.
$(Ax12)$ $((\varphi \rightarrow \psi) \equiv ((\varphi \vee \psi) \equiv \psi))$.
$(Ax13)$ $((\varphi \leftarrow \psi) \equiv (\psi \rightarrow \varphi))$.
$(Bx1)$ $((\forall x\, \varphi) \equiv \varphi)$, *if* x *is not free in* φ.
$(Bx2)$ $((\varphi \vee (\forall x\, \psi)) \equiv (\forall x\, (\varphi \vee \psi)))$, *if* x *is not free in* φ.
$(Bx3)$ $(((\forall x\, \varphi) \wedge (\forall x\, \psi)) \equiv (\forall x\, (\varphi \wedge \psi)))$.

$(Bx4)$ $((\forall x\,\varphi) \rightarrow \varphi[x := t])$, *if t is free for x in φ.*
$(Bx5)$ $((\exists x\,\varphi) \equiv (\neg\,(\forall x\,(\neg\varphi))))$.
$(Bx6)$ $(x = x)$.
$(Bx7)$ $((x = t) \rightarrow (\varphi \equiv \varphi[x := t]))$, *if t is free for x in φ.*

The axioms of $\mathsf{DS}(\mathcal{L})$ can be divided into two groups, namely, $(Ax1)$-$(Ax13)$ and $(Bx1)$-$(Bx7)$. Axioms $(Ax1)$, $(Ax2)$, and $(Ax3)$ define that equivalence is associative, commutative, and has identity element *true*. Similarly, axioms $(Ax4)$, $(Ax5)$, and $(Ax6)$ define that disjunction is associative, commutative, and has identity element *false*. Disjunction is idempotent by Axiom $(Ax7)$ and distributes over equivalence by Axiom $(Ax8)$. The remaining axioms $(Ax9)$-$(Ax13)$ present axiomatic definitions for the connectives in the propositional fragment of $\mathsf{DS}(\mathcal{L})$. Axiom $(Bx1)$ states that a universal quantifier on variable x can be omitted whenever the formula it quantifies has no free occurrences of x. Axiom $(Bx2)$ states that disjunction distributes over universal quantification whenever there is no variable capture, while Axiom $(Bx3)$ states that conjunction and universal quantification commute. By Axiom $(Bx4)$, it is possible to particularize any universal quantification with a term t whenever the variables in t are not captured by the substitution. Finally, Axiom $(Bx5)$ is an axiomatic definition for existential quantification.

Note that by having $\{P_0, P_1, \ldots\} \subseteq \mathcal{P}$ in Definition 2, propositions over propositional variables $\{p_0, p_1, \ldots\}$ can be represented as atomic formulas in $\mathcal{T}(\emptyset, \emptyset, \mathcal{P})$ via the mapping $p_i \mapsto P_i$. With this embedding, axioms $(Ax1)$-$(Ax13)$ characterize the set $\{true, false, \equiv, \vee\}$ as a complete collection of connectives for the propositional fragment of $\mathsf{DS}(\mathcal{L})$. Likewise, $\{true, false, \equiv, \vee, \forall\}$ is a complete collection of connectives for $\mathsf{DS}(\mathcal{L})$.

Definition 5. *Let $x \in \mathcal{X}$, $P \in \mathcal{P}$ with $ar(P) = 0$, and $\varphi, \psi, \tau \in \mathcal{T}(\mathcal{X}, \mathcal{F}, \mathcal{P})$. The* inference rules *of* $\mathsf{DS}(\mathcal{L})$ *are:*

$$\frac{\psi \qquad (\psi \equiv \varphi)}{\varphi}\ \textsc{Equanimity} \qquad \frac{(\psi \equiv \tau)}{(\varphi[P := \psi] \equiv \varphi[P := \tau])}\ \textsc{Leibniz} \qquad \frac{\varphi}{(\forall x\,\varphi)}\ \textsc{Generalization}.$$

Rules Equanimity and Leibniz allow for symbolic manipulation based on equality by substitution of ‘equals for equals’. Rule Generalization is the usual first-order rule stating that universally quantifying any theorem results in a theorem. The assumption about the *infinite* collection of constant predicate symbols in $\mathsf{DS}(\mathcal{L})$ is key for the Rule Leibniz to work when substituting formulas in any given formula. Another important fact regarding Rule Leibniz is that from it some meta-properties can be proved with almost no effort: (i) any substitution instance of a tautology (i.e., of a theorem in the propositional fragment of $\mathsf{DS}(\mathcal{L})$) is a theorem of $\mathsf{DS}(\mathcal{L})$ and (ii) the collection of theorems of $\mathsf{DS}(\mathcal{L})$ is closed under formula substitution.

‘Proofs’ in the Dijkstra-Scholten calculational style are not strict in the sense of a formal system. Instead, they are sequences of formulas related, mainly, by equivalence. This approach takes advantage of the transitive properties of the connectives to obtain compact proof calculations.

Definition 6. *Let Γ be a collection of formulas of* $\mathsf{DS}(\mathcal{L})$*. A* derivation from Γ *in* $\mathsf{DS}(\mathcal{L})$ *is a non-empty finite sequence of formulas $\varphi_0, \varphi_1, \ldots, \varphi_n$ of* $\mathsf{DS}(\mathcal{L})$ *satisfying, for any $0 < k \leq n$, $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} (\varphi_{k-1} \equiv \varphi_k)$.*

The connection between a derivation and a proof is made precise in Proposition 1.

Proposition 1. *[10] Let Γ be a collection of formulas of* $\mathsf{DS}(\mathcal{L})$ *and* $\varphi_0, \varphi_1, \ldots, \varphi_n$ *be a derivation in* $\mathsf{DS}(\mathcal{L})$ *from* Γ. *It holds that* $\Gamma \vdash_{\mathsf{DS}(\mathcal{L})} (\varphi_0 \equiv \varphi_n)$.

It is important to note that any proof in the formal system $\mathsf{DS}(\mathcal{L})$ is a derivation in $\mathsf{DS}(\mathcal{L})$ but a derivation is *not* necessarily a proof. Consider, for instance, the sequence "*false*, *false*" which is a derivation because Boolean equivalence is reflexive, but this sequence is not a proof because *false* is not a theorem. The key fact about proofs in a formal system is that every formula in a proof is a theorem, while this is not necessarily the case in a derivation. There are other types of derivations where implication or consequence can be combined with equivalence (see [10] for details).

In practice, derivations are not written directly as a sequence of formulas but as a bi-dimensional arrangement of formulas and text explaining each derivation step.

Remark 1. A derivation $\varphi_0, \varphi_1, \ldots, \varphi_n$ from Γ in $\mathsf{DS}(\mathcal{L})$ is usually written as:

$$
\begin{array}{ll}
 & \varphi_0 \\
\equiv & \langle \text{“explanation}_0\text{”} \rangle \\
 & \varphi_1 \\
\vdots & \langle \ldots \rangle \\
 & \varphi_{n-1} \\
\equiv & \langle \text{“explanation}_{n-1}\text{”} \rangle \\
 & \varphi_n
\end{array}
$$

in which "explanation$_i$" is a text describing why $\Gamma \vdash_{\mathsf{DS}} (\varphi_i \equiv \varphi_{i+1})$.

Finally, the Dijkstra-Scholten logic proposes an alternative notation for writing quantified formulas. The main idea is that proof verification and derivation in such a syntax becomes simpler thanks to the resemblance between, for example, the notation of a (finite) quantification and the operational semantics of repetitive constructs in an imperative programming language.

Remark 2. Let $x \in \mathcal{X}$ and φ, ψ be formulas of $\mathsf{DS}(\mathcal{L})$.

- The expression $(\forall x \mid \psi : \varphi)$ is syntactic sugar for $(\forall x\,(\psi \to \varphi))$; in particular, $(\forall x \mid \mathit{true} : \varphi)$ can be written as $(\forall x \mid: \varphi)$.
- The expression $(\exists x \mid \psi : \varphi)$ is syntactic sugar for $(\exists x\,(\psi \wedge \varphi))$; in particular, $(\exists x \mid \mathit{true} : \varphi)$ can be written as $(\exists x \mid: \varphi)$.

In the formulas $(\forall x \mid \psi : \varphi)$ and $(\exists x \mid \psi : \varphi)$, ψ is called the *range* and φ the *subject* of the quantification.

3 An Axiomatic Set Theory

This section presents Set, a Zermelo-Franekel first-order system in the language $\mathcal{L}_{\mathrm{Set}} = (\mathcal{X}, \mathcal{F}, \mathcal{P})$, that results from extending $\mathsf{DS}(\mathcal{L}_{\mathrm{Set}})$ with axioms for sets. In $\mathcal{L}_{\mathrm{Set}}$, the infinitely many variables $\mathcal{X}$ range over elements in the domain of discourse, $\mathcal{F}$ contains the constant $\varnothing$ and the unary symbols $\bigcup, \mathbb{P}$, and the only predicate symbol in $\mathcal{P}$ is the binary symbol $\in$. Intuitively, the function symbols represent the empty set, generalized union, and the power set; the predicate symbol $\in$ represents membership. The axioms of Set in Definition 7 include axiomatic definitions for all symbols in $\mathcal{F}$, meaning that $\in$ is a complete connective for Set (i.e., the entire language of set theory can be built from the membership predicate symbol). Symbols not in $\mathcal{L}_{\mathrm{Set}}$ such as the binary function symbols $\cup$ and $\cap$ denoting union and intersection, respectively, and the binary predicate symbol $\subseteq$ denoting inclusion can be added by means of the usual axiomatic definitions. Some examples will be given at the end of the section.

Note that in $\mathcal{L}_{\mathrm{Set}}$ there is no mention of the common 'curly braces' notation $\{\_ \mid \_\}$ used for identifying collections, because this notation can also be seen as an abbreviation just like $\varnothing$ or $\subseteq$. Technically, $\{\_ \mid \_\}$ is a binary meta-symbol used as a term-forming operator that can be defined with the *definite description* operator ι, for any variable $x \in \mathcal{X}$ and formula $\varphi \in \mathcal{T}(\mathcal{X}, \mathcal{F}, \mathcal{P})$, as follows:

$$\{x \mid \varphi(x)\} \equiv (\iota y \mid: (\forall x \mid: x \in y \equiv \varphi(x)))\,.$$

It can be shown, although it is beyond the scope of this paper, that $\{x \mid \varphi\}$ identifies a unique element. Therefore, Set allows the 'curly braces' notation to be used as an abbreviation for a unique element in the domain of discourse. The reader is referred to [13, p. 126] for details on the definite description operator and its properties.

The notion of univalent formula is needed before introducing the axioms of Set. A formula $\varphi \in \mathcal{T}(\mathcal{X}, \mathcal{F}, \mathcal{P})$ is *univalent* iff $(\forall x, y, z \mid \varphi(x, z) \wedge \varphi(y, z) : x = y)$. Intuitively, if φ is univalent and $\varphi(x, z)$ is true, then x is the only element that makes $\varphi(\_, z)$ true.

Definition 7. *Let $\varphi, \psi \in \mathcal{T}(\mathcal{X}, \mathcal{F}, \mathcal{P})$ be such that φ has one free variable and ψ is univalent. The axioms of* Set *are given by the following axiom schemata:*

$(Cx1)$ $(\forall x \mid: x = \varnothing \equiv (\forall y \mid: \neg y \in x))$.
$(Cx2)$ $(\forall x, y \mid: x = y \equiv (\forall u \mid: u \in x \equiv u \in y))$.
$(Cx3)$ $(\forall x, y, z \mid: x = \{y, z\} \equiv (\forall u \mid: (u \in x \equiv u = y \vee u = z)))$.
$(Cx4)$ $(\forall x, y \mid: y = \{u \in x \mid \varphi(u)\} \equiv (\forall u \mid: u \in y \equiv u \in x \wedge \varphi(u)))$.
$(Cx5)$ $(\forall x, y \mid: y = \bigcup x \equiv (\forall u \mid: u \in y \equiv (\exists z \mid z \in x : u \in z)))$.
$(Cx6)$ $(\forall x, y \mid: y = \mathbb{P}x \equiv (\forall u \mid: u \in y \equiv (\forall z \mid z \in u : z \in x)))$.
$(Cx7)$ $(\forall x, y \mid: y = \psi[x] \equiv (\forall u \mid: u \in y \equiv (\exists z \mid z \in x : \psi(u, z))))$.
$(Cx8)$ $(\exists x \mid \varnothing \in x : (\forall y \mid y \in x : y \cup \{y\} \in x))$.

The *axiom of existence* $(Cx1)$ serves two purposes: first, it states the existence of an unique set without elements, namely, the empty set; second, it is a 'definitional extension' for the function symbol $\varnothing$, which is the name assigned to the empty set. Note that, by identifying 'the' set without elements with $\varnothing$, there is the need to prove that such a set is unique (this proof is left to the reader as a routine exercise after covering this section). The *axiom of extensionality* $(Cx2)$ characterizes equality: two elements are equal whenever they have the same elements. The *axiom of pairing* $(Cx3)$ states the existence of an element having two given elements. The *axiom schema of separation* $(Cx4)$, which represents as many axioms as formulas φ with exactly one variable are, states how an element can be obtained from other element by selecting exactly those elements that satisfy a given formula. The *axiom of union* ($Cx5$) and the *axiom of power* ($Cx6$), respectively, define generalized union and the power element construction. The *axiom schema of replacement* ($Cx7$) uses an univalent formula to define an element $\psi[x]$ comprising precisely those elements witnessing the satisfaction of $\psi(\_, z)$, for each $z \in x$. Finally, the *axiom of infinity* ($Cx8$) introduces the existence of (at least) one *inductive set*: (i) $\varnothing$ belongs to this set; and (ii) if x belongs to this set, then $x \cup \{x\}$ (i.e., its *successor set*) is also one of its members. It is easy to see that such sets must necessarily have infinitely many elements starting from $\varnothing$, the successor of $\varnothing$, and so on. Also note that in Definition 7 the only axiom schemata are ($Cx4$) and ($Cx7$) because they are parametric on given formulas.

In general, these axioms are similar to the ones usually studied in graduate-level axiomatic set theory courses. A contribution of Set is a rewrite of the axioms in the notation of Dijkstra-Scholten. However, as illustrated in Section 4, the main contribution of Set is that it enables an undergraduate proof-based course on set theory using simple algebraic manipulation.

One important cornerstone of any axiomatic set theory, including Set, is the distinction between elements that are 'well-behaved' and those that are not. More precisely, axiomatic set theory distinguishes the elements that can be called a *set* from others that are not, namely, the broader concept of a *class*. Technically, a class is any collection, but a set is a more refined version of a class: a set is a collection that can be *identified* by only using the axioms in Definition 7. For instance, $\varnothing$ and $\{\varnothing\}$ are sets because of axioms ($Cx1$) and ($Cx8$). Theorem 1 presents a fundamental theorem of Set, with a proof à la Dijkstra-Scholten, and identifies a class that is not a set: the collection of all sets.

Theorem 1. *There exists no universal set.*

Proof. Towards a contradiction, assume such a set V exists. Thus, $\vdash_{\mathsf{Set}} (\forall x \mid : x \in V)$. Consider the set $S = \{x \in V \mid x \notin x\}$, in which $x \notin x$ abbreviates $\neg x \in x$:

$$
\begin{aligned}
& S \in S \\
\equiv\ & \langle \text{ definition of } S \ \rangle \\
& S \in \{x \in V \mid x \notin x\} \\
\equiv\ & \langle \text{ axiom of separation } (Cx4) \ \rangle \\
& S \in V \wedge S \notin S \\
\equiv\ & \langle\ V \text{ is a universal set } \rangle \\
& \mathit{true} \wedge S \notin S \\
\equiv\ & \langle \text{ propositional logic } \rangle \\
& S \notin S.
\end{aligned}
$$

That is, $\vdash_{\mathsf{Set}} S \in S \equiv S \notin S$, which is a contradiction. Therefore, V cannot exist. □

As mentioned at the beginning of this section, other usual function and predicate symbols can be added to Set by means of definitional extensions. Some of these symbols are included in Definition 8.

Definition 8. *The following axioms define pairing, binary union and intersection, difference, and inclusion:*

($Cx10$) $(\forall x, y, z \mid: x = (y, z) \equiv (\forall u \mid: u \in x \equiv u = \{y\} \vee u = \{y, z\}))$.
($Cx11$) $(\forall x, y, z \mid: x = y \cup z \equiv (\forall u \mid: u \in x \equiv u \in y \vee u \in z))$.
($Cx12$) $(\forall x, y, z \mid: x = y \cap z \equiv (\forall u \mid: u \in x \equiv u \in y \wedge u \in z))$.
($Cx13$) $(\forall x, y, z \mid: x = y \setminus z \equiv (\forall u \mid: u \in x \equiv u \in y \wedge u \notin z))$.
($Cx14$) $(\forall x, y, z \mid: x = y \times z \equiv (\forall u \mid: u \in x \equiv (\exists v, w \mid v \in y \wedge w \in z : u = (v, w))))$.
($Cx15$) $(\forall x, y \mid: x \subseteq y \equiv (\forall u \mid u \in x : u \in y))$.

Other operations such as the generalized Cartesian product and generalized intersections, and the axiom of choice can be defined similarly in the syntax of Set.

4 Calculational Proofs for the Classroom

This section presents some notorious features of Set that have been identified, mainly, by teaching a 16-weeks undergraduate set theory course. It is important to note that most of the students in such a course are in their sophomore year and have had very little exposure to mathematical logic. First, as it is often the case in set theory, computation of operations between sets depends heavily on the axiom of extensionality ($Cx2$). Since Set is based mainly on Boolean equivalence, algebraic manipulations are simple to grasp and can help a student in discovering proofs. Second, the precise language required for writing formulas exposes their logical structure, thus making it possible in a proof to transform one formula into another in a clean way. Furthermore, in the Dijkstra-Scholten style a student can deal symbolically with the parts of a proof argument that have to do exclusively with propositional logic, usually hidden in a rhetorical argument. Finally, Set helps to identify the logical structure of a theorem text and to anticipate relevant lemmas for its proof.

4.1 Algebraic Exploration

One of the advantages of Set is that it facilitates the computation of operations between elements in the domain of discourse. The axiom of extensionality ($Cx2$) is key in situations when the goal is to transform a formula $x \in A$ to another formula $x \in B$. In this setting, A is a set whose definition is known, while B is a set to be found.

Example 1. The goal is to simplify $\bigcup\{\varnothing, \{\varnothing\}\}$:

$x \in \cup\{\varnothing, \{\varnothing\}\}$
$\equiv$ ⟨ axiom of union ($Cx5$) ⟩
$(\exists y \mid y \in \{\varnothing, \{\varnothing\}\} : x \in y)$
$\equiv$ ⟨ axiom of pair ($Cx3$) ⟩
$(\exists y \mid y = \varnothing \vee y = \{\varnothing\} : x \in y)$
$\equiv$ ⟨ syntactic sugar for existential quantification; propositional logic ⟩
$(\exists y \mid: (y = \varnothing \wedge x \in y) \vee (y = \{\varnothing\} \wedge x \in y))$
$\equiv$ ⟨ axiom of empty set ($Cx1$): no element belongs in the empty set ⟩
$(\exists y \mid: \mathit{false} \vee (y = \{\varnothing\} \wedge x \in y))$

$\equiv$ ⟨ syntactic sugar for existential quantification; propositional logic ⟩
$(\exists y \mid y = \{\varnothing\} : x \in y)$
$\equiv$ ⟨ exactly one element satisfies the range ⟩
$x \in \{\varnothing\}$.

Thus, $\bigcup\{\varnothing, \{\varnothing\}\} = \{\varnothing\}$. □

Another illustrative example is reasoning with function composition. If f and g are two functions, then $g \circ f$ is defined by:

$$(\forall x, y \mid: (x, y) \in g \circ f \equiv (\exists z \mid: (x, z) \in f \wedge (z, y) \in g)) .$$

As in [4], the expression $\langle f_i \mid i \in I\rangle$ denotes the function f with domain I. For example, the function $f(x) = x^2$ with domain $[0, 1]$ can be represented as $\left\langle x^2 \mid x \in [0, 1]\right\rangle$. The formula $(u, v) \in \left\langle x^2 \mid x \in [0, 1]\right\rangle$ means that $v = u^2$ and that $u \in [0, 1]$. An example in [4] is to calculate $\left\langle \sqrt{x} \mid x > 0\right\rangle \circ \left\langle x^2 + 1 \mid x \in \mathbb{R}\right\rangle$. For that purpose, the authors use an extra theorem to first determine the domain of the composition and then proceed to compute it. As presented by Example 2, the preliminary theorem is not required because the domain of the composition can be obtained simultaneously with the proof.

Example 2. The goal is to compute $\left\langle \sqrt{x} \mid x > 0\right\rangle \circ \left\langle x^2 - 1 \mid x \in \mathbb{R}\right\rangle$:

$(u, v) \in \left\langle \sqrt{x} \mid x > 0\right\rangle \circ \left\langle x^2 - 1 \mid x \in \mathbb{R}\right\rangle$
$\equiv$ ⟨ definition of function composition ⟩
$\left(\exists z \mid: (u, z) \in \left\langle x^2 - 1 \mid x \in \mathbb{R}\right\rangle \wedge (z, v) \in \left\langle \sqrt{x} \mid x > 0\right\rangle\right)$
$\equiv$ ⟨ syntactic sugar ; $\langle \_ \mid \_ \rangle$ notation ⟩
$\left(\exists z \mid: z = u^2 - 1 \wedge u \in \mathbb{R} \wedge v = \sqrt{z} \wedge z > 0\right)$
$\equiv$ ⟨ Axiom ($Bx3$); Axiom ($Bx1$) ⟩
$u \in \mathbb{R} \wedge \left(\exists z \mid: z = u^2 - 1 \wedge v = \sqrt{z} \wedge z > 0\right)$
$\equiv$ ⟨ only one element satisfies the range ⟩
$u \in \mathbb{R} \wedge v = \sqrt{u^2 - 1} \wedge u^2 - 1 > 0$
$\equiv$ ⟨ syntactic sugar ; $\langle \_ \mid \_ \rangle$ notation ⟩
$(u, v) \in \left\langle \sqrt{x^2 - 1} \mid x^2 - 1 > 0\right\rangle$.

Therefore, $\left\langle \sqrt{x} \mid x > 0\right\rangle \circ \left\langle x^2 - 1 \mid x \in \mathbb{R}\right\rangle = \left\langle \sqrt{x^2 - 1} \mid x^2 > 1\right\rangle$. □

4.2 Discovery of Logical Structure

The calculative style requires writing the propositions in a very precise language that ultimately reveals their logical structure. This makes it possible to carry out the required transformations from one proposition to another in a proof more transparently than when using a language that has not been designed for such a purpose. In an axiomatic theory, arguments in a proof are expected to be very precise, leaving aside – as far as possible –, colloquial ones. For example, it can be shown that the Cartesian product of two sets is empty iff one of its factors is empty. Of course, nobody doubts this fact but, in order to proceed formally, it requires a proof.

Example 3. Prove $\vdash_{\mathsf{Set}} (\forall x, y \mid: x \times y = \varnothing \equiv (x = \varnothing \vee y = \varnothing))$.

$$
\begin{array}{ll}
 & x \times y = \varnothing \\
\equiv & \langle \text{ axiom of empty set } (Cx1) \rangle \\
 & (\forall u, v \mid: \neg(u, v) \in x \times y) \\
\equiv & \langle \text{ axiom of Cartesian product } (Cx14) \rangle \\
 & (\forall u, v \mid: \neg (u \in x \wedge v \in y)) \\
\equiv & \langle \text{ propositional logic: De Morgan's law } \rangle \\
 & (\forall u, v \mid: u \notin x \vee v \notin y) \\
\equiv & \langle \text{ first-order logic } \rangle \\
 & (\forall u \mid: u \notin x) \vee (\forall v \mid: v \notin y) \\
\equiv & \langle \text{ axiom of empty set } (Cx1) \rangle \\
 & x = \varnothing \vee y = \varnothing .
\end{array}
$$

Therefore, $(\forall x, y \mid: x \times y = \varnothing \equiv (x = \varnothing \vee y = \varnothing))$ is a theorem of Set. □

One of the objectives of teaching a set theory course is to develop the ability to properly write all arguments of a proof in natural language. This task is usually a complex one for students, especially when the arguments are related with Boolean reasoning. This is because such a reasoning is used implicitly in proofs, leaving the bitter feeling that the argument is correct but without clarifying the reasons. In argumentative proofs, in general, the reasoning rests more on aesthetic matters rather than on logical ones. Within the Set formal system all arguments are made explicit, which helps to improve the clarity and forcefulness of the proofs without resorting to the elegance and skill in the use of the natural language.

In Example 4, it is proved that a function is invertible iff it is one to one. This is a case of use of appropriate language when searching for a proof. The predicate $\mathrm{fun}(f)$ stands for "f is a function", $\mathrm{inv}(f)$ for "f is invertible", and $\mathrm{oto}(f)$ for "f is one to one". Symbolically,

$$
\begin{array}{l}
(\forall f \mid: \mathrm{fun}(f) \equiv (\forall x, y, z \mid (x, y) \in f \wedge (x, z) \in f : y = z)). \\
\left(\forall f \mid: \mathrm{inv}(f) \equiv \mathrm{fun}(f^{-1})\right). \\
(\forall f \mid: \mathrm{oto}(f) \equiv (\forall x, y, z \mid (x, z) \in f \wedge (y, z) \in f : x = y)).
\end{array}
$$

Example 4. Prove $\vdash_{\mathsf{Set}} (\forall x \mid: \mathrm{inv}(f) \equiv \mathrm{oto}(f))$.

$$
\begin{array}{ll}
 & \mathrm{inv}(f) \\
\equiv & \langle \text{ definition of invertible function } \rangle \\
 & \mathrm{fun}(f^{-1}) \\
\equiv & \langle \text{ definition of function } \rangle \\
 & \left(\forall x, y, z \mid (x, y) \in f^{-1} \wedge (x, z) \in f^{-1} : y = z\right)
\end{array}
$$

$\equiv$ ⟨ definition of inverse relation ⟩
$(\forall x, y, z \mid (y, x) \in f \wedge (z, x) \in f : y = z)$
$\equiv$ ⟨ definition of one to one function ⟩
$\mathrm{oto}(f)$.

□

Example 5 presents a proof in Set that natural numbers with the usual order $<$ are a well-ordered set. The set $\mathbb{N}$ of natural numbers is the smallest inductive set and membership is a strictly linear relation. Well-ordering of $\mathbb{N}$ means that all non-empty subsets of $\mathbb{N}$ have a first (or $<$-minimal) element. Consider the set of least numbers $x_{\min}$ of a given set $x \subseteq \mathbb{N}$:

$$(\forall x \mid x \subseteq \mathbb{N} : x_{\min} = \{y \in x \mid (\forall z \mid z \in x : z \leq y)\}) .$$

Of course, if $x_{\min}$ is not empty, then it is unitary. Next consider the unary predicate $\mathrm{wo}_<$ defined as follows:

$$(\forall x \mid x \subseteq \mathbb{N} : \mathrm{wo}_<(x) \equiv (\forall y \mid y \subseteq x : y \neq \varnothing \rightarrow y_{\min} \neq \varnothing)) .$$

Note that $\mathrm{wo}_<(\mathbb{N})$ means that the set of natural numbers is well-ordered. The definition of well-order can equivalently be written, thanks to $\vdash_{\mathsf{DS}(\mathcal{L})} (\varphi \rightarrow \psi) \equiv (\neg\psi \rightarrow \neg\varphi)$, as:

$$(\forall x \mid x \subseteq \mathbb{N} : \mathrm{wo}_<(x) \equiv (\forall y \mid y \subseteq x : y_{\min} = \varnothing \rightarrow y = \varnothing)) .$$

In addition, Example 5 uses a form of derivation in which logical implication is allowed to relate a deduction step. Such a sequence is called *relaxed derivation* and the reader is referred to [10] for its definition and properties.

Example 5. Prove $\vdash_{\mathsf{Set}} (\forall x \mid x \subseteq \mathbb{N} : x_{\min} = \varnothing \rightarrow x = \varnothing)$.

$x_{\min} = \varnothing$
$\equiv$ ⟨ axiom of empty set $(Cx1)$ ⟩
$(\forall n \mid: n \notin x_{\min})$
$\equiv$ ⟨ definition of $x_{\min}$ ⟩
$(\forall n \mid: n \notin x \vee (\exists k \mid k \in x : k < n))$
$\equiv$ ⟨ first-order logic ⟩
$(\forall n \mid: (\forall k \mid k < n : k \notin x) \rightarrow n \notin x)$
$\rightarrow$ ⟨ induction principle for natural numbers ⟩
$(\forall n \mid: n \notin x)$
$\equiv$ ⟨ axiom of empty set $(Cx1)$ ⟩
$x = \varnothing$.

□

4.3 Proof Structure and Organization

The calculative style can be used to anticipate auxiliary lemmas. Indeed, this is the case in some induction theorems, such as the proof of commutativity of natural number addition. Addition of natural numbers is as a function $+ : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$ defined à la Peano by:

$$(\forall m \mid m \in \mathbb{N} : +(0, m) = m),$$
$$(\forall m, n \mid m \in \mathbb{N} \wedge n \in \mathbb{N} : +(m, +(n, 1)) = +(+(m, n), 1)).$$

The goal is to prove that natural number addition is commutative, i.e, that

$$(\forall m, n \mid m \in \mathbb{N} \wedge n \in \mathbb{N} : +(m, n) = +(n, m))\,.$$

In [4, Thm.4.4, p.53]), the proof of this fact is hard to follow because there is an auxiliary induction proof inside the main induction proof:

> We prove that every $n \in \mathbb{N}$ commutes, by induction on n. To show that 0 commutes, it suffices to show that $0 + m = m$ for all m. .... Clearly $0 + 0 = 0$, and if $0 + m = m$, then $0 + (m + 1) = (0 + m) + 1 = m + 1$. So the claim follows by induction (on m). Let us assume that n commutes, an let us show that $n + 1$ commutes. We prove, by induction on m, that $m + (n + 1) = (n + 1) + m$ for all $n \in \mathbb{N}$. ...(proof follows)...

In [2, Sec.13, p.50], the following is the proof of the same property:

> The proof that addition is commutative ... is a little tricky; a straightforward attack might fail. The trick is to prove, by induction on n, that (i) $0 + n = n$ and (ii) $m^+ + n = (m + n)^+$ and then to prove the desire commutativity equation by induction on m, via (i) y (ii).

It will be better for a student to identify beforehand some lemmas needed for the proof.

Example 6. Note that the theorem statement has the form $(\forall n \mid n \in \mathbb{N} : q(n))$, where

$$q(n) \equiv (\forall m \mid m \in \mathbb{N} : p(m, n))\,.$$

Some lemmas can be found from this formulation:

$$\begin{array}{ll} & (\forall n \mid n \in \mathbb{N} : q(n)) \\ \leftarrow & \langle \text{ induction principle } \rangle \\ & q(0) \wedge (\forall n \mid q(n) : q(n + 1))\,. \end{array}$$

Therefore, the goal now is to first prove $(\forall m \mid m \in \mathbb{N} : p(m, 0))$ and then, under hypothesis $(\forall m \mid m \in \mathbb{N} : p(m, n))$, prove $(\forall m \mid m \in \mathbb{N} : p(m, n + 1))$.

5 Classroom Experience

With the calculational style there is an opportunity to read differently theorem statements and to rethink theorem formulations in the teaching of mathematics. Sometimes the logical complexity of the statements exceeds the capacity of the students and the complexity of informal speech can blur the simplicity of logical structure. It is gratifying to have a language to explore the meaning of the statements by cleansing them from literary linguistic figures. The style of Dijkstra and Scholten does precisely that. Reading and writing proofs in the style of Dijkstra and Scholten reveals structural propositions that are hidden in the nooks and crannies of literary languages.

Sophomore students in the axiomatic set theory class, in which Set is taught, are familiarized quickly with the formal system $\mathsf{DS}(\mathcal{L})$ before beginning the study of set theory. Without being logic experts and after learning these logic rudiments, they develop an ability to translate the statements of the theorems and proofs contained in a textbook such as [4]. Students are bound by their teacher to do the inverse process of reading and writing in literary language what they have written symbolically.

Later on, students in the class begin to propose their own proofs, aside from those found in the textbook. At a later stage, they see the need to introduce new symbols to the language (e.g., a predicate symbol) in the quest to incorporate what is said of objects as part of the symbolic discourse. The instructors are very strict in requiring students to write sentences in symbolic language and precise spelling is enforced. Finally, students have no difficulty in accepting the need to introduce new sets to the theory in order to 'package' information and to be able to argue on a more abstract level: 'set-set' rather than 'set-element' (e.g., instead of referring to the minimum of a set x, they need to refer to the subset $x_{\mathrm{m}in}$ of x).

6 Concluding Remarks

Most textbooks on set theory avoid dealing with formal logic directly. They include not only the introductory textbooks such as [2], but also graduate-level ones such as [4,6,7]. It is believed among the mathematical community, that detailed formal proof involves a large amount of trivial details that would make a standard hundred-page mathematical book run thousands of pages. However, this is hardly the case with the approach of Dijkstra and Scholten, as shown with Set. First-order logic systems such as natural deduction or sequent calculus are suitable for mechanical reasoning but not for human reasoning. It is fair to say that the substitution of 'equals for equals' is ultimately the reason why the calculational approach is practical for human reasoning. Important meta-mathematical aspects of $\mathsf{DS}(\mathcal{L})$ that have been omitted in this paper, such as the deduction, soundness, and completeness results can be found in [12,10]. Moreover, a sequent-like formalization of $\mathsf{DS}(\mathcal{L})$ can be found in [12], as well as an explanation about the relationship between the propositional fragment of $\mathsf{DS}(\mathcal{L})$ and the celebrated rewrite-based decision procedure for Boolean rings of J. Hsiang [5].

Perhaps, the only textbook that takes a self-contained approach to set theory, in the sense that includes all tools needed from mathematical logic, is [13] by Tourlakis. His work is situated between opposite poles: on the one hand, it works within the theory, that is, uses the tools and the axioms for the sole purpose of proving theorems. On the other hand, it takes the entire theory as an object of study and "from the outside" answers questions about its power and reliability. Although the use of formal reasoning as a tool to calculate proofs is recommended, not many proofs in the book are obtained in this way. Furthermore, the author of [13] states that "we do not have to be that formal always, nor can we afford to be so when our arguments get more involved. We will frequently relax the proof style to shorten proofs. This relaxing will invariably use shorthand tools such as English text, class terms, and a judicious omission of (proof) details." Its level of exposition is designed to fit a spectrum of mathematical sophistication, well beyond the reach of most inexperienced undergraduate students. Set encourages the use of formal proofs when teaching set theory.

The results presented in this paper are part of a larger effort to formally study and mechanize topics in mathematics and computer science with the algebraic approach of E. W. Dijkstra and C. S. Scholten. The next step is to mechanize Set in rewriting logic [9]. There is already experience by some of the authors in mechanizing logical systems in rewriting logic [12,11]. Finally, there is also interest in exploring the formalization and use of the Dijkstra-Scholten style in other branches of mathematics and computer science such as topology, number theory, and finite model theory.

Acknowledgments. The authors would like to thank the members of the two-term seminar for fruitful discussions on these ideas, and Jorge Finke and the anonymous referees for comments that helped to improve the paper.

References

1. E. W. Dijkstra and C. S. Scholten. *Predicate Calculus and Program Semantics*. Texts and monographs in computer science. Springer-Verlag, New York, 1990.
2. P. R. Halmos. *Naive Set Theory*. Undergraduate texts in mathematics. Springer-Verlag, New York, 1974.
3. R. E. Hodel. *An Introduction to Mathematical Logic*. Dover Publications, Inc., Mineola, New York, 2013. OCLC: 892599098.
4. K. Hrbacek and T. J. Jech. *Introduction to Set Theory*. Number 220 in Monographs and textbooks in pure and applied mathematics. M. Dekker, New York, 3rd ed., rev. and expanded edition, 1999.
5. J. Hsiang. Refutational theorem proving using term-rewriting systems. *Artificial Intelligence*, 25(3):255–300, Mar. 1985.
6. T. J. Jech. *Set Theory*. Number 79 in Pure and applied mathematics, a series of monographs and textbooks. Academic Press, New York, 1978.
7. K. Kunen. *Set Theory*. Number 34 in Studies in logic. College Publications, London, revised edition edition, Oct. 2013. OCLC: 915461876.
8. J. Meseguer. General Logics. In *Logic Colloquium '87: Proceedings*, volume 129 of *Studies in Logic and the Foundations of Mathematics*, pages 275–330. Elsevier, Granada, Spain, first edition edition, Aug. 1989.
9. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, Apr. 1992.
10. C. Rocha. The Formal System of Dijkstra and Scholten. In N. Martí-Oliet, P. C. Ölveczky, and C. Talcott, editors, *Logic, Rewriting, and Concurrency*, volume 9200, pages 580–597. Springer International Publishing, Cham, 2015.
11. C. Rocha and J. Meseguer. A Rewriting Decision Procedure for Dijkstra-Scholten's Syllogistic Logic with Complements. *Revista Colombiana de Computación*, 8(2):101–130, Dec. 2007.

12. C. Rocha and J. Meseguer. Theorem Proving Modulo Based on Boolean Equational Procedures. In R. Berghammer, B. Möller, and G. Struth, editors, *Relations and Kleene Algebra in Computer Science*, volume 4988, pages 337–351. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
13. G. J. Tourlakis. *Lectures in Logic and Set Theory*. Number 82-83 in Cambridge studies in advanced mathematics. Cambridge University Press, Cambridge, UK ; New York, 2003.