Semantics of OpenMath and MathML3

Michael Kohlhase, Florian Rabe

Computer Science, Jacobs University Bremen
{m.kohlhase,f.rabe}@jacobs-university.de

Abstract. Even though OpenMath has been around for more than 10 years, there is still confusion about the "semantics of OpenMath". As the upcoming MathML3 recommendation will semantically base Content MathML on OpenMath Objects, this question becomes more pressing.

One source of confusions about OpenMath semantics is that it is given on two levels: a very weak algebraic semantics for expression trees, which is extended by considering mathematical properties in content dictionaries that interpret the meaning of (constant) symbols. While this two-leveled way to interpret objects is well-understood in logic, it has not been spelt out rigorously for OpenMath.

In this paper we look at the semantics of OpenMath from a foundational point of view and reconcile this "semantics" with the foundations of mathematics established in the early 20th century; the traditional way of assigning meaning to mathematical objects.

Note to Reviewers: *We are aware that this paper is over the 12-page limit of the OpenMath 2009 workshop. But we have not had time to cut it down without endangering readability yet. Given time we believe that we can summarize in 12 pages, so that this should not be seen as a problem for acceptance. But as the workshop only has electronic proceedings we are not sure whether reaching the 12 page limit is worth the effort.*

1 Introduction

MathML2 [ABC+03] and OpenMath2 [BCC+04] are standards for the representation and communication of mathematical objects. Even though they have been around for more than 10 years, there is still confusion about the "semantics of OpenMath". As the upcoming MathML3 recommendation will semantically base Content MathML on OpenMath Objects, this question becomes more pressing.

1.1 OpenMath and MathML

MathML comes in two parts: *presentation* MathML provides XML-based layout primitives for the traditional two-dimensional notation of mathematical formulae and *content* MathML, which focuses on encoding the meaning of objects rather than visual representations to allow the free exchange of mathematical objects between software systems and human beings. OpenMath has the same goals as content MathML, but was developed by a different community with slightly different intuitions. Both representation formats represent mathematical objects as expression trees. Content MathML tries to cover all of school and engineering mathematics (the "K-14" fragment) in a representation format intuitive to mathematicians, and OpenMath concentrates on an extensible framework built on a minimal structural core language with a well-defined extension mechanism. Where MathML supplies more than a dozen elements for special constructions, OpenMath only supplies concepts for function application (`OMA`), binding constructions (`OMBIND`), and attributions (`OMATTR`). Where MathML provides close to 100 elements for the K-14 fragment, OpenMath gets by with only an `OMS` element that identifies symbols by pointing to declarations in an open-ended set of Content Dictionaries.

An OpenMath Content Dictionary (CD) is a document that declares names (OpenMath "symbols") for basic mathematical concepts and objects. CDs act as the unique points of reference for OpenMath symbols (via `OMS` elements) and thus supply a notion of context that situates and disambiguates OpenMath expression trees. To maximize modularity and reuse, a CD typically contains a relatively small collection of definitions for closely related concepts. The OpenMath Society maintains a large set of public CDs, including CDs for all pre-defined symbols in MathML2. There is a process for contributing privately developed CDs to the OpenMath Society repository to facilitate discovery and reuse. OpenMath does not require CDs be publicly available, though in most situations the goals of semantic markup will be best served by referencing public CDs available to all user agents.

To avoid fragmentation and smooth out interoperability obstacles, effort is currently under way to align OpenMath and MathML semantically. To remedy the lack of regularity and specified meaning in MathML, content MathML was extended by concepts like binding structures and full semantic annotations from OpenMath and a structurally regular subset of the extended content MathML was identified that is isomorphic to OpenMath objects. This subset is called **strict content MathML** to contrast it to full content MathML that is seen to strike a more pragmatic balance between regularity and human readability. Full content MathML borrows the semantics from strict MathML by a mapping specified in the MathML3 specification [ABC$^+$08][1] that defines the meaning of non-strict (**pragmatic**) MathML expressions in terms of strict MathML equivalents. Strict Content MathML in turn obtains its meaning by being an encoding of OpenMath Objects.

In this situation, the "meaning of OpenMath (Objects)" obtains a completely new significance; especially when OpenMath still receives evaluations like

[1] By the time of publication we expect a much enhanced and "feature-complete" working draft of the MathML3 spec to have been published

> *On the other hand the paper leaves me unsatisfied, and even irritated. It is frustrating to know that the* MathML*3 and* OpenMath*3 standards still will be meaningless from a semantic point of view. [. . .] will not lead to a standard for mathematical expressions where those expressions have a proper semantics.* anonymous referee for [DK09]

The aim of this paper is to clarify the status of semantics in OpenMath (and thus content MathML3) and in particular counter sentiments like the one above. We see the reason for this "misunderstanding" in a presentational gap between how mathematical objects and theories are conventionally given a meaning and the way OpenMath answers the question. In the rest of this section, we will briefly recap the established foundations (of meaning in) Mathematics and the way OpenMath establishes meaning. Based on this, we will bridge the differences and clarify gray areas in a formal semantic analysis in section 2. Section 4 concludes the paper.

1.2 Foundations of Mathematics

The question of what the meaning of mathematical expressions and theories might be is usually by methods from **Logic**, a scientific field at the intersection of philosophy and mathematics concerned with the study of the concepts **proposition** and **truth** and the reasoning about them.

The age-old question about the meaning of language in general and mathematics in particular turned into the "*Grundlagenkrise*" of mathematics by the discovery of paradoxa, i.e., contradictions, in what is called **naive set theory** in retrospect. Naive set theory was the implicitly assumed foundation of mathematics at the time, Cantor's "Grundlagen" ([Can83]) from 1883 being the most influential contribution. The best known paradoxon was found by Russell in 1901 ([Rus01]). Peano had noticed a similar one in 1897.

In response to this, mathematicians have developed several — sometimes alternative, sometimes complementary — foundations (i.e. specific logics picked as a starting point of mathematics) that can replace naive set theory. This happened over several decades as an evolutionary creative process. But it did not culminate in a commonly accepted solution. Rather, it led to profound and sometimes fierce debates on what mathematics is. The personal quarrel between Hilbert and Brouwer, which was partially fuelled by these debates, is an almost tragic example. From this evolution emerged two major classes of foundations: **axiomatic set theory** and **type theory**.

The basic idea of **axiomatic set theory** is that there is a universe of sets, and any mathematical object ever introduced is a set. The sets are related via the binary relations of equality and membership. For example $m \in M$ is used to say that the set m is a member of the set M. Depending on context, M is regarded as a property of m or as a structuring concept. To talk about sets, equality, membership, and propositions are used. The basic propositions are of the form $m = m'$ and $m \in M$. Composed propositions are built up from the basic ones. Typically, (at least) first-order logic (FOL) is used as the language of composed propositions: FOL uses propositions such as $F \wedge G$ and $\forall x.F(x)$ denoting "F and G are true" and "for all (sets) x, F is true about x". Then a limited collection of propositions (the **axioms**) is chosen as fundamental truths. These are chosen very carefully to prevent contradictions and to obtain a minimal set of axioms. Based on the axioms, **proofs** are used to single out the true propositions. A proof consists of a sequence of steps that derive one true proposition from other true propositions starting with the axioms. In this way the whole of mathematics is developed, and for every proposition, truth is defined by whether it has a proof.

Both set theory and type theory have led to **numerous specific foundations** of mathematics. Zermelo-Fraenkel set theory, based on [Zer08,Fra22], is most commonly in use today. Other variants are von Neumann-Bernays-Gödel set theory, based on [vN25,Ber37,Göd40], which is important for category theory, and Tarski-Groethendieck set theory, based on [Tar38,Bou64]. The first type theory was Russells's ramified theory of types ([Rus08]). And in their Principia ([WR13]), Whitehead and Russell gave one of the most influential foundations of mathematics. Church's simple theory of types, also called higher-order logic, ([Chu40]) is the most-used type theory today. Important other type theories are typically organized in the lambda cube ([Bar91]) and include dependent type theory ([ML74,**?**]), System F ([Gir71,Rey74]), and the calculus of constructions ([CH88]). Most of these foundations have further variants, such as Zermelo-Fraenkel set theory with or without the Axiom of Choice or type theory with or without product types.

Hilbert's formalistic program, set forth in his second problem ([Hil00]) and various texts from the 1920s, e.g., [Hil26], called for the reduction of all mathematics to a set of axioms and a consistency proof for these axioms using only finitary means. Since proofs are built up from the axioms, such a reduction would yield all true propositions by systematically searching all proofs. In 1930, Gödel established two negative results ([Göd31]), which as von Neumann recognized first showed that the goal of Hilbert's program is unreachable.

The first one roughly says that no foundation of mathematics can be found that defines the truth of all propositions in an algorithmic way. The second one says that no foundation can prove its own consistency. Gödel worked in the Principia, which were the foundation mainly in use at the time, but the results extend to all foundations beyond a certain level of expressivity. This is the major reason why no foundation has won the endorsement of mathematicians as a whole and why there will not be a final answer which foundation of mathematics is the best. Since no perfect foundation exists, the personal preferences and the characteristics of a problem lead to different choices of foundation.

In any case, most mathematicians today accept a mathematical object or theory as "meaningful", iff it can (in principle) be formalized in one of the foundations, most notably set theory. Incidentally, most logics in use have a set-theoretic (i.e. foundational) semantics that makes them acceptable to mathematicians in this sense.

1.3 The Meaning of OpenMath

The OpenMath standard actually gives two answers to the question about the meaning of OpenMath expressions. The first one comes from the fact that OpenMath is intended as a communication standard between mathematical software systems: OpenMath envisions communication via *phrasebooks* (see [BCC$^+$04, chapter 1]): Each mathematical software system S is equipped with an OpenMath phrasebook that converts OpenMath expressions from and to the internal representations of the system S. In this "system communication view", the meaning is built into the phrasebooks that (purport to) understand the expression, and the meaning is whatever S (after conversion by the phrasebook) makes it to be. Clearly, this view of meaning is not very helpful, and taken in the radical simplicity we have formulated it here is not an adequate account. After all, the purpose of the OpenMath standard is to is to synchronize what the system-specific representations of objects, so that communication between systems is meaning-preserving. To attain this goal, OpenMath does two things:

1. It defines "OpenMath objects" which acts as the model for encodings of mathematical formulae. OpenMath objects are essentially labeled trees, modulo α-conversion for binding structures and flattening for nested semantic annotations. The OpenMath standard considers OpenMath objects as primary citizens and views the "OpenMath XML encoding" as just an incidental design choice for an XML-based markup language. In fact OpenMath specifies another encoding: the "binary encoding" designed to be more space efficient at the cost of being less human-readable.
2. Rather than appealing to mathematical intuition, OpenMath stipulates that phrasebooks should be informed by (mathematical properties in) content dictionaries.

   > *It is the OpenMath Content Dictionaries which actually hold the meanings of the objects being transmitted. For example if application A is talking to application B, and sends, say, an equation involving multiplication of matrices, then A and B must agree on what a matrix is, and on what matrix multiplication is, and even on what constitutes an equation. All this information is held within some Content Dictionaries which both applications agree upon. [. . .] The primary use of Content Dictionaries is thought to be for designers of Phrasebooks, the programs which translate between the OpenMath mathematical object and the corresponding (often internal) structure of the particular application in question.* [BCC$^+$04, section 4.1]

Even if this is not spelt[2] out in the OpenMath2 standard the algebra $\mathcal{O}$ of OpenMath objects as an can be interpreted[3] as an (initial) model for encodings

[2] In particular the "compliance chapter" does not mention mathematical properties in CDs at all.

[3] To the best of our knowledge, this "act of interpretation" has never been backed by a formal mathematical study; which is what prompted the work reported in this paper.

of mathematical formulae. Note that since $\mathcal{O}$ is initial it is essentially unique and identifies (in the sense of "declares to be the same") fewer objects than any other model. As a consequence two mathematical objects must be identical if their OpenMath representations are, but not the other way around.

While this can be seen as a failure of OpenMath to supply semantics ("OpenMath is only syntax"), we see it as an expression of the OpenMath representational philosophy expressed in

> *OpenMath objects do not specify any computational behaviour, they merely represent mathematical expressions. Part of the OpenMath philosophy is to leave it to the application to decide what it does with an object once it has received it. OpenMath is not a query or programming language. Because of this, OpenMath does not prescribe a way of forcing "evaluation" or "simplification" of objects like* $2 + 3$ *or* $\sin(\pi)$*. Thus, the same object* $2 + 3$ *could be transformed to* 5 *by a computer algebra system, or displayed as* $2 + 3$ *by a typesetting tool.* [BCC$^+$04, section 1.5]

In this sense the initial algebra semantics of OpenMath objects is intentionally weak to make the OpenMath format ontologically unconstrained and thus universally applicable. It basically represents the accepted design choice of representing objects as formulae. Any further (meaning-giving) properties of an object o are relegated to the content dictionaries referenced in o, where they can be specified formally (as "Formal Mathematical Properties" as `FMP` elements containing OpenMath objects) or informally (as "Commented Mathematical Properties" as `CMP` elements containing text). Thus the precision of OpenMath as a representation language can be adapted by supplying CDs to range from fully formal (by providing CDs based on some logical system) to fully informal (where CDs are essentially empty except for declaring symbols).

In the next section, we will formally develop the initial algebra semantics of OpenMath objects, and then in section 3 extend it to take mathematical properties in CDs into account, thus showing that the interpretation above can indeed be made mathematical and be reconciled with the notion of meaning in foundations of mathematics.

2 An Algebraic Semantics for OpenMath Objects

We will now define a an algebraic semantic semantics for OpenMath objects building on ideas from [BBK04]. The difference to the situation there (giving a semantics for the simply typed λ calculus with a type of Booleans) is that OpenMath allows n-ary function application (rather than binary) arbitrary binding symbols (rather than just λ-abstraction), and arbitrary attributions (rather than just simple types), but only assumes α-conversion (rather than $\alpha\beta\eta$ conversion).

2.1 Syntax

We start out by fixing an abstract syntax of "OM objects", which we will relate to OpenMath objects in Section 2.3. We will call the objects according to

Definition 4 *abstract* OM Objects when we want to distinguish from the *standard* OpenMath objects defined in the OpenMath2 standard [BCC+04, section 2].

Definition 1 (Symbols and Variables). In all of the following, we will assume the existence of two disjoint, countably infinite sets: a set *Symbols* of **symbols** and a set *Variables* of **variables**. Furthermore, we assume a set $\mathit{Keys} \subseteq \mathit{Symbols}$ of **keys**.

As usual in formal languages we are a little more careful about the variables we use in the construction, vocabularies and contexts as defined above helps us do this.

Definition 2 (OM Vocabulary). An OM **vocabulary** is a set of symbols. For every OM vocabulary T, we denote by $\mathit{Symbols}(T) := \mathit{Symbols} \cap T$ the set of symbols of T and by $\mathit{Keys}(T) := \mathit{Keys} \cap T$ the set of symbols of T.

Definition 3 (OM Context). An OM **context** C is a finite list of variables. We will write $(x_1, \ldots, x_n)$ for such lists and use $+$ for concatenation of lists.

Definition 4 (OM Objects). Let T be an OM vocabulary. The set $O(T, C)$ of **OM-objects** over T in context C is the smallest set closed under the following operations

1. if $s \in \mathit{Symbols}(T) \setminus \mathit{Keys}(T)$, then $\mathbb{S}(s) \in O(T, C)$,
2. if $x \in C$, then $\mathbb{V}(x) \in O(T, C)$,
3. if $f, o_1, \ldots, o_n \in O(T, C)$, then $\mathbb{A}(f, o_1, \ldots, o_n) \in O(T, C)$,
4. if $b \in O(T, C)$, $X_1, \ldots, X_n \in \mathit{AttVar}(T, C)$, and $o \in O(T, C')$ where $C' = C + (\mathit{varname}(X_1), \ldots, \mathit{varname}(X_{l(\sigma)})))$, then $\mathbb{B}(b, X_1, \ldots, X_n, o) \in O(T, C)$,
5. if $o \in O(T, C)$, $k \in \mathit{Keys}(T)$, and $v \in O(T, C)$, then $\mathbb{K}(o|k := v) \in O(T, C)$,

where the name of an attributed variable is defined by $\mathit{varname}(\mathbb{K}(o'|k := v)) = \mathit{varname}(o')$ and $\mathit{varname}(\mathbb{V}(x)) = x$.

Here **attributed variables** are defined by: $o \in \mathit{AttVar}(T, C)$ if a $o = \mathbb{V}(x)$ for some $x \in C$ or $o = \mathbb{K}(o'|k := v) \in O(T, C)$ for some $o' \in \mathit{AttVar}(T, C)$. We call OM objects in the empty context **ground**.

Note that in contrast to the OpenMath2 standard we only consider "unary" attributions which associate an object with a single key/value pair. This allows us to build the "flattening of attributions" into the abstract representation of OM Objects. We can regain the syntactic structure of OpenMath2 objects by introducing n-ary attributions as an abbreviation for nested attributions: $\mathbb{K}(o|k_1 := v_1, \ldots, k_n := v_n) = \mathbb{K}(\mathbb{K}(o|k_1 := v_1)|k_2 := v_2, \ldots, k_n := v_n)$ for $n \geq 2$. With this trick[4] we have fully covered the requirement of "attribution flattening equivalence" required in the OpenMath standard.

[4] In fact we propose to follow this path in the next version of the OpenMath standard as it simplifies the presentation. Note that we are only talking about abstract OpenMath objects, not their XML or binary encodings, where n-ary attributions make sense for notational convenience.

Let us fortify our intuition with an example; we choose a binding object, since this is the problematic case. We will use this as a running example in the following

Example 1. The untyped universal quantification $\forall x.x = x$ is represented as $\mathbf{U} = \mathbb{B}(\mathbb{S}(\forall), (\mathbb{V}(x)), \boxed{x = x})$[5], where $\forall$ is a symbol. To show the interaction of attribution and binding, we use typed function represented as a λ-abstraction: the typed quantification $\lambda x : T.x$ is represented as $\mathbf{L} = \mathbb{B}(\mathbb{S}(\lambda), (\mathbb{K}(\mathbb{V}(x)|\tau := \boxed{T})), \mathbb{V}(x))$, where τ is a key symbol (i.e. a symbol with role "semantic-attribution").

The use of attributed variables in binders can lead to a somewhat awkward notations when accessing the keys and attributions present in abstract binding objects. Therefore, we use the auxiliary definition of binding signatures. Intuitively, an OpenMath binding has binding signature σ if it binds $l(\sigma)$ variables where the i-th variable has $d^i(\sigma)$ attributions.

Definition 5 (Binding Signature). A **binding signature** σ consists of

- a positive natural number $l(\sigma)$ (the **length** of σ),
- natural numbers $d^1(\sigma),\ \ldots,\ d^n(\sigma)$ (the **depth of σ at** i).

We denote by $\overline{\sigma}$ the set of pairs $(i, j) \in \mathbb{N} \times \mathbb{N}$ where $1 \leq i \leq l(\sigma)$ and $1 \leq j \leq d^i(\sigma)$.

If σ is a binding signature with length n, $b \in O(T, C)$, $K : \overline{\sigma} \to \mathit{Keys}(T)$, and $V : \overline{\sigma} \to O(T, C)$, and $o \in O(T, C + (x_1, \ldots, x_n))$, then we write

$$\mathbb{B}(b\,[x_1, \ldots, x_n|K := V].o) \quad \text{for} \quad \mathbb{B}(b, (X_1, \ldots, X_n), o) \in O(T, C)$$

where $X_i = \mathbb{K}(\mathbb{V}(x_i)|K(i, 1) := V(i, 1), \ldots, K(i, d^i(\sigma)) := V(i, d^i(\sigma)))$.

Clearly, every OM object of the form $\mathbb{B}(b, (X_1, \ldots, X_n), o)$ can be written uniquely as an expression of the form $\mathbb{B}(b\,[(x_1, \ldots, x_n)|K := V].o)$, and we will use the latter notation in the future.

Example 2 (Continuing Example 1). In the abbreviated syntax $\forall x.x = x$ is represented as $\mathbf{U} := \mathbb{B}(\mathbb{S}(\forall)\,[x\,|\,\varnothing := \varnothing].\boxed{x = x})$ and $\lambda x : T.x = x$ as $\mathbf{L} := \mathbb{B}(\mathbb{S}(\lambda)\,[x|K := V].\mathbb{V}(x))$, where

- $l(\sigma) = 1$ and $d^1(\sigma) = 1$, and therefore $\overline{\sigma} = \{(1, 1)\}$
- $K = \{(1, 1) \mapsto \tau\}$ and $V = \{(1, 1) \mapsto \boxed{T}\}$

Definition 6 (Substitution). For $o \in O(T, (x_1, \ldots, x_n))$, the substitution function that maps $(o_1, \ldots, o_n)$ to the object with all variables x_i substituted with o_i is denoted by $\mathit{Subs}(o)$.

Definition 7 (α-Equality). Two objects are said to be **α-equal** iff they arise from one another by renaming bound variables. $\equiv_\alpha$ denotes the induced equivalence relation, and $[o]_\alpha$ denotes the equivalence class of o.

[5] Here and throughout the paper we will use boxed mathematical formulae to gloss OpenMath objects (encoded, abstract, or standard; we assume that this distinction is either meaningless or clear from the context).

2.2 Semantics

In the following, we will use use the notation $\Lambda x \in A.f(x)$. for the set-theoretical function defined by $\{(x, f(x)) : x \in A\}$. A may be omitted if it is clear from the context. We also write B^A for the set of functions from A to B.

Definition 8 (OM Algebra). Let T be an OM vocabulary. An **OM-algebra** A over T consists of

1. a set $U := U^A$ called the **universe of discourse**
2. a family of sets $R^A_n \subseteq U^{(U^n)}$ for $n \geq 1$,
3. an element $s^A \in U$ for every $s \in \mathit{Symbols}(T) \setminus \mathit{Keys}(T)$,
4. a family of mappings $@^A_n : U \times U^n \to U$ for $n \geq 1$,
5. a family of mappings $\beta^A_K : U \times U^{\overline{\sigma}} \times R_{l(\sigma)} \to U$ for mappings $K : \overline{\sigma} \to \mathit{Keys}(T)$ and binding signatures σ,
6. a family of mappings $\alpha^A_k : U \times U \to U$ for every $k \in \mathit{Keys}(T)$.

Whereas s^A, $@^A$, β^A, and α^A are intended to interpret symbols, applications, bindings, and attributions, respectively in a relatively standard fashion, the sets R^A_n are special. Syntactically, binders are operators that take terms with free variables as arguments. It is well-understood in higher-order logic and type theory that this can be modeled by operations taking functions as arguments. Here, since OpenMath binders can take arbitrarily many variables, we need to interpret binders as operations taking n-ary functions on the universe as arguments.

Since we can always write a binder like $\mathbb{B}(b\,[x\,|\,\varnothing := \varnothing].x)$ (for the empty binding signature), R^A_1 should at least contain the identity function. However, the whole set $U^{(U^n)}$ is too big since only some of these functions actually arise from the interpretation of terms with free variables. Since the interpretation of these terms depends on A itself, we permit an arbitrary set R^A_n here and leave it to Def. 10 to sort out when an OM-algebra is well-defined.

Definition 9 (Assignment). Let A be an OM-Algebra over T, and let C be an OM-context. An A-**assignment** φ for C is a mapping from C to U^A. We denote by $\varphi, [x/o]$ the assignment for $C + (x)$ that maps x to o and agrees with φ for all other variables.

Definition 10 (Interpretation). Let A be an OM-Algebra over T, and let φ be an A-assignment for a context C. The **interpretation** $[\![o]\!]^A_\varphi$ of $o \in O(T, C)$ in A under φ is defined as follows:

1. $[\![\mathbb{S}(s)]\!]^A_\varphi = s^A$,
2. $[\![\mathbb{V}(x)]\!]^A_\varphi = \varphi(x)$,
3. $[\![\mathbb{A}(f, o_1, \ldots, o_n)]\!]^A_\varphi = @^A_n([\![f]\!]^A_\varphi, ([\![o_1]\!]^A_\varphi, \ldots, [\![o_n]\!]^A_\varphi))$,
4. $[\![\mathbb{B}(b\,[x_1, \ldots, x_n|K := V].o)]\!]^A_\varphi = \beta^A_K([\![b]\!]^A_\varphi, \mathcal{V}, \mathcal{F})$ where
   (a) σ is the binding signature of the binding (which must have length n),
   (b) $\mathcal{V} = \Lambda p \in \overline{\sigma}.[\![V(p)]\!]^A_\varphi$,
   (c) $\mathcal{F} = \Lambda u \in (U^A)^n.[\![o]\!]^A_{\varphi,[x_1/u_1,\ldots,x_n/u_n]}$

5. $\llbracket \mathbb{K}(o|k := v)\rrbracket^A_\varphi = \alpha^A_k(o, v)$.

Whether the case for bindings is well-defined, depends on the sets R^A_n. We call A **well-defined** if $\Lambda u \in (U^A)^n.\llbracket o\rrbracket^I_{\varphi,[x_1/u_1,\ldots,x_n/u_n]} \in R^A_n$ for all C, n, $o \in O(T, C)$, and φ.

Example 3 (Continuing Example 2). To interpret the first example from Example 2, we use an OM Algebra A with

1. $U := \mathbb{N} \cup \{\mathsf{q}, \mathsf{e}, \mathsf{l}, \mathsf{a}, \mathsf{t}, \mathsf{f}\}$
2. $R^A_n = U^{(U^n)}$,
3. $\forall^A := \mathsf{q}$, $=^A := \mathsf{e}$, $\lambda^A := \mathsf{l}$, and $\tau^A := \mathsf{a}$
4. $@^A_2(\mathsf{e}, u, v) = \mathsf{t}$ if $u = v$ and $@^A_n(u, u_1, \ldots, u_n) = \mathsf{f}$ otherwise.
5. $\beta^A_\varnothing(\mathsf{q}, \varnothing, \mathcal{F}) = \mathsf{t}$ if $\mathcal{F}(u) = \mathsf{t}$ for all $u \in \mathbb{N}$; and $\beta^A_K(u, (x_1, \ldots, x_n), \mathcal{F}) = \mathsf{f}$ otherwise,
6. $\alpha^A_\tau : (u, v) = u$.

If we want to evaluate $\boxed{\forall x.x = x}$ in this algebra, recall that $\overline{\sigma} = \varnothing$ and thus $\mathcal{V} = \Lambda p \in \overline{\sigma}.\llbracket \varnothing(p)\rrbracket^A_\varnothing = \varnothing$, so we have

$$\llbracket \mathbf{U}\rrbracket^A_\varnothing = \llbracket \mathbb{B}(\mathbb{S}(\forall)\,[x|\varnothing := \varnothing].\boxed{x = x})\rrbracket^A_\varnothing = \beta^A_\varnothing(\mathsf{q}, \varnothing, \mathcal{F})$$

where $\mathcal{F} = \Lambda u \in U^A.\llbracket \boxed{x = x}\rrbracket^A_{[x/u]}$. So $\llbracket \mathbf{U}\rrbracket^A_\varnothing = \mathsf{t}$, iff $\mathcal{F}(u) = \mathsf{t}$ for all $u \in \mathbb{N}$. But observe that we have $\mathcal{F}(u) = \llbracket \mathbb{A}(=, \mathbb{V}(x), \mathbb{V}(x))\rrbracket^A_{[x/u]} = @^A_2(\mathsf{e}, u, u) = \mathsf{t}$ by definition, and thus $\llbracket \mathbf{U}\rrbracket^A_\varnothing = \mathsf{t}$ as expected.

Extending A to an interpretation of the λ-binder is more complicated because we have to commit to a type theory. For example, we can extend U^A so that it contains all function sets that can be formed from the natural numbers, i.e., $\mathbb{N}^\mathbb{N}$ $\mathbb{N}^{(\mathbb{N}^\mathbb{N})}$, $(\mathbb{N}^\mathbb{N})^\mathbb{N}$ and so on, as well as the functions they contain. (For this to be useful, we should also extend our vocabulary with symbols ι and $\rightarrow$ and put $\iota^A = \mathbb{N}$, $\rightarrow^A = \mathsf{p}$ and interpret $@^A_2(\mathsf{p}, u, v)$ as the set of functions from v to u if u and v are sets and as f otherwise.) Then for $\overline{\sigma} = \{(1, 1)\}$, $K = \{(1, 1) \mapsto \tau\}$, we can put $\beta^A_K(\mathsf{l}, \mathcal{V}, \mathcal{F})$ to be the function $\Lambda u \in \mathcal{V}(1, 1).\mathcal{F}(u)$. Furthermore, we put $@^A_2(f, u) = f(u)$ whenever function application is defined.

Then we can interpret $\boxed{\lambda x : T.x}$ as follows. We have $\llbracket \mathbf{L}\rrbracket^A_\varnothing = \llbracket \mathbb{B}(\mathbb{S}(\lambda)\,[x|K := V].\mathbb{V}(x))\rrbracket^A_\varnothing = \beta^A_\varnothing(\mathsf{l}, \mathcal{V}, \mathcal{F})$ where

- $\mathcal{V} = \Lambda p \in \{(1, 1)\}.\llbracket V(p)\rrbracket^A_\varnothing = \Lambda p \in \{(1, 1)\}.\llbracket \boxed{T}\rrbracket^A_\varnothing$,
- $\mathcal{F} = \Lambda u \in U.\llbracket \mathbb{V}(x)\rrbracket^A_{[x/u]} = \Lambda u \in U.u$

And thus, we evaluate $\beta^A_\varnothing(\mathsf{l}, \mathcal{V}, \mathcal{F})$ as the identity function on $\llbracket \boxed{T}\rrbracket^A_\varnothing$ as expected.

A simple induction over the construction of OpenMath objects in Definition 4 using the respective clauses in Definition 10 gives us an OpenMath version of the well-known

Lemma 1 (Substitution Value Lemma). $\llbracket [x/o']o\rrbracket^A_\varphi = \llbracket o\rrbracket^\varphi_{a,[x/\llbracket o'\rrbracket^A_\varphi]}$

Proof. This is shown by induction on o.

This in turn can be specialized in the usual way to obtain:

Corollary 1 (Soundness of α-Equality). *If $o \equiv_\alpha o'$ then $[\![o]\!]^A_\varphi = [\![o']\!]^A_\varphi$.*

So we have shown that OM algebras form a model class for OpenMath objects. We will now show that they characterize them up to isomorphism. For that we need to consider initial models, which will function as canonical representatives in this model class.

Definition 11 (Free OM Algebra). Let T be an OM vocabulary. Let $\overline{Subs(o)}$ abbreviate the function $\Lambda([o_1]_\alpha, \ldots, [o_n]_\alpha) \in U^n.[Subs(o)(o_1, \ldots, o_n)]_\alpha$. Then the **free OM algebra** $I := I(T)$ **over** T is defined as follows.

1. $U^I = O(T, \emptyset)/_{\equiv_\alpha}$, i.e. the quotient set of the ground OpenMath objects modulo α-conversion.
2. R^I_n is the set of functions $\overline{Subs(o)}$ for $o \in O(T, (x_1, \ldots, x_n))$,
3. $s^I = [\mathbb{S}(s)]_\alpha$,
4. $@^I_n([f]_\alpha, ([o_1]_\alpha, \ldots, [o_n]_\alpha)) = [\mathbb{A}(f, o_1, \ldots, o_n)]_\alpha$,
5. for a binding signature σ: $\beta^I_K([b]_\alpha, \mathcal{V}, \mathcal{F}) = [\mathbb{B}(b\,[x_1, \ldots, x_n \,|\, K := V].o)]_\alpha$ where
   - $V = \Lambda p \in \overline{\sigma}.v_p$ for some $v_p \in \mathcal{V}(p)$,
   - $o \in O(T, (x_1, \ldots, x_n))$ is some object such that $\overline{Subs(o)} = \mathcal{F}$.
6. $\alpha^I_k([o]_\alpha, [v]_\alpha) = [\mathbb{K}(o|k := v)]_\alpha$.

Lemma 2. *$I(T)$ is well-defined.*

Proof. We need to show several well-definedness conditions.

Well-definedness of $\overline{Subs(o)}$ Substituting ground α-equivalent objects preserves α-equivalence. This follows from the definition of α-equivalence.

Well-definedness of $@^I_n$ If $f \equiv_\alpha f', o_1 \equiv_\alpha o'_1, \ldots, o_n \equiv_\alpha o'_n$, then $\mathbb{A}(f, o_1, \ldots, o_n) \equiv_\alpha \mathbb{A}(f', o'_1, \ldots, o'_n)$. This follows directly from the definition of α-equivalence.

Well-definedness of β^I_K If $b \equiv_\alpha b'$, $v(p) \equiv_\alpha v'(p)$ for all $p \in \overline{\sigma}$, then there exists an $o \in O(T, (x_1, \ldots, x_n))$ such that $\overline{Subs(o)} = \mathcal{F}$, and for two such o, o' we have that

$$\mathbb{B}(b\,[x_1, \ldots, x_n|K := \Lambda p.v_p].o) \equiv_\alpha \mathbb{B}(b'\,[x_1, \ldots, x_n|K := \Lambda p.v'_p].o').$$

The existence follows from the definition of R^I_n. The α-equivalence holds because non-α-equivalent objects induce non-α-equivalent substitution functions.

Well-definedness of α^I_k If $o \equiv'_\alpha$ and $v \equiv_\alpha v'$, then $\mathbb{K}(o|k := v) \equiv_\alpha \mathbb{K}(o'|k := v')$. This follows directly from the definition of α-equivalence.

Lemma 3. *Let T be an OM algebra. Then for every $o \in O(T, ())$, we have $[\![o]\!]^{I(T)} = [o]_\alpha$.*

Proof. This is proved by a straightforward induction on the structure of o.

Lemma 4 ($I(T)$ is initial). *Let A be an OM algebra over T, and let $I := I(T)$. Then there is a (unique) mapping $h : U^I \to U^A$ satisfying $h([\![o]\!]^{I(T)}) = [\![o]\!]^{\varphi}$.*

Proof. This follows directly from Cor. 1 and Lem. 3.

Corollary 2 (Completeness of α-Equality). *If $[\![o]\!]^A_\varphi = [\![o']\!]^A_\varphi$ for all OM algebras A, then $o \equiv_\alpha o'$.*

Proof. This follows from Lem. 3 by putting $A := I(T)$.

2.3 OpenMath Objects with Uninterpreted Symbols

The semantics discussed so far was based on the abstract notion of OM Vocabularies. To arrive at a semantics of OpenMath objects we need to relate this to OpenMath CDs.

The OpenMath2 standard introduces "abstract content dictionaries" to abstract from the concrete XML encoding of content dictionaries. According to [BCC$^+$04, section 4.2], (abstract) CDs have a **CD name**, a **CD base URI**, and contain **symbol definitions**, which in turn consist (among others) of a **symbol name**, a **symbol role** (one of "binder", "attribution", "semantic-attribution", "application", "constant", and "error"), and a set of **mathematical properties**.

Definition 12 (OpenMath Symbols). We say that a CD C **declares** an OpenMath symbol (n, c, u, r), iff the CD base of C is u, the CD name of C is c, and C has a symbol definition with symbol name n and symbol role r (note that the role can be undefined as it is optional). We define the set *Symbols* to be the set of symbols declared by some OpenMath CD and the set *Keys* to be those with symbol role "semantic-attribution".

There are three differences between abstract OM Objects and standard OpenMath objects; all three are related to symbols and keys:

1. We do not take keys to be abstract OM objects by themselves (see clause 1 in the definition above). We claim that that there are no mathematically meaningful situations where keys can appear except in attributions. This design decision should not be perceived as a serious impediment for our semantics, since keys can be added analogously to the treatment below at the cost of adding an additional case everywhere.
2. The OpenMath2 [BCC$^+$04] "role system", poses some additional restrictions on where symbols can occur, but not enough to simplify our construction of binding signatures below. Therefore, we disregard it here and refer the reader to [RK09a] for details and an extended role system proposal that would.
3. We do not consider attributions with symbols that are not in *Keys*, in particular symbols with roles "attribution" which are intended by the OpenMath2 standard for just this purpose. However the standard states

> *This form of attribution may be ignored by an application, so should be used for information which does not change the meaning of the attributed OpenMath object.* [BCC$^+$04, clause 2.1.4.*ii*]

and therefore it necessary to disregard these attributions in the construction of a semantics for OpenMath. In the mapping from standard OpenMath objects to abstract ones, we strip attributions with non-*Keys* symbols.

This allows us to define the meaning of an OpenMath object. As we are not taking mathematical properties in CDs into account, we will think of these symbols as uninterpreted, therefore we will call it the "algebraic meaning".

Definition 13 (Algebraic Meaning). Let o be an OpenMath object, then we call the set of symbols such that s where $\mathbb{S}(s)$ occurs in o the **OM vocabulary induced by** o.

If $o \in O(T, ())$ is a ground OM Object, T its induced vocabulary, and A an OM-algebra over T, then the **algebraic meaning of** o **in** A is $[\![o]\!]^A$ and the **algebraic meaning of** o is $[\![o]\!]^{I(T)}$.

Note that the algebraic meaning of an abstract OpenMath object in the sense of Definition 4 is just its (standard) OpenMath object.

As discussed in the introduction, the algebraic semantics only gives us a rather weak and syntactic concept of meaning of the OpenMath language. To understand the full meaning of OpenMath objects we need to take CDs into account, which we do in the next section.

3 OpenMath Models

If we want to understand mathematical properties in OpenMath content dictionaries, we need to have a notion of "truth" — after all the properties are assumed to hold. Furthermore, we need to take into account the mathematical properties themselves. In OpenMath there are two kinds of mathematical properties: "commented mathematical properties" (encoded as `CMP` elements which contain mathematical vernacular) and "formal mathematical properties" (encoded as `FMP` elements that contain XML encodings of OpenMath objects). We are going to concentrate on the latter in this paper since they provide more structure. This is no loss of generality, given the assumption in mathematical practice that any rigorously stated property can be fully formalized given enough resources. For for the purposes of this paper we will just assume that we have access to an oracle that translates all commented mathematical properties into formal one, which we handle.

3.1 Theories and Satisfaction

As formal mathematical properties are expressed as OpenMath objects, we will need to build the required notion of "truth" into an OM vocabulary, which is rather simple.

Definition 14 (Logic). An OM vocabulary L with distinguished symbols $\top$ and $=$ is called a logic.

In OpenMath CDs, (formal) mathematical properties are expressed as statements in some foundational logical system, thus the OM Objects representing them will in general contain symbols from the foundation and the CD itself. For instance, the `arith1` CD [CDa04] contains an `FMP` with the object $\boxed{\forall a, b.a + b = b + a}$ to express commutativity of addition. The symbols $\boxed{\forall}$ and $\boxed{=}$ are from the vocabulary of the foundational system and the symbol $\boxed{+}$ is from the CD itself.

We will treat OpenMath content dictionaries as logical theories, which are determined by their vocabularies and axioms, and model them using institutions (see [Rab08] for an introduction to both).

Definition 15 (Theory). Let L be a logic and T an OM vocabulary. We denote the union of L and T by $L \cup T$. An **OM theory** Θ for L is a pair $(T, \mathit{Axioms}(\Theta))$ where $\mathit{Axioms}(\Theta) \subseteq O(L \cup T, ())$. An OM algebra over Θ is an OM algebra over $L \cup T$.

In this setting we can define OM models as those algebras that respect equality and in which the axioms hold.

Definition 16 (Model). Let Θ be an OM theory for L. An OM algebra over Θ is a **model** of Θ if

- for all V, $o, o' \in O(T, V)$ and φ, we have that $[\![\mathbb{A}(=, o, o')]\!]^A_\varphi = [\![\top]\!]^A$ iff $[\![o]\!]^A_\varphi = [\![o']\!]^A_{\varphi'}$,
- for all $F \in \mathit{Axioms}(\Theta)$, we have that $[\![F]\!]^A = [\![\top]\!]^A$.

The **Model Class** $\mathcal{M}(C)$ **of** Θ is the set of OM Models of Θ.

This gives us the standard notions of satisfaction and semantic entailment.

Definition 17 (Satisfaction). Let L be a logic, Θ be an OM theory for L, $o \in O(\Theta, V)$, M an OM model of Θ, and φ an assignment for V into M. Then we say that M **satisfies** o **under** φ (which we denote as $M, \varphi \models o$), iff $[\![o]\!]^M_\varphi = [\![\top]\!]^M_\varphi$. We write $M \models o$ if $M, \varphi \models o$ holds for all assignments φ and say that o is *valid* in M.

Definition 18 (Entailment). Let Θ be an OM theory and o a ground object. Then we say that Θ **entails** o ($\Theta \models o$), iff $M \models o$ for all $M \in \mathcal{M}(\Theta)$.

3.2 The Meaning of OpenMath CDs and Objects

Note that the definitions above are still abstract in the sense that they refer to OM vocabularies, and OM theories, and not OpenMath CDs. So as in section 2.3 we have to relate abstract OM objects to standard ones and in particular to answer the question: what is the theory of a content dictionary? The

OpenMath2 standard leaves this information under-defined, so we propose an interpretation that allows us to define an adequate notion of mathematical semantics[6]. Note that OpenMath CDs need not be self-contained, i.e. their `FMP`s can contain symbols that are neither introduced in the CD nor from the foundational system. Of course, these symbols (and thus the CDs that introduce them) should have an effect on the meaning of the symbols described by the `FMP`, so they need to be taken into account; naturally this process must be iterated until fixed point has been reached.

Definition 19 (CD Import). Let C be an OpenMath content dictionary, then we say that C **imports** D, iff $C \neq D$ and some `FMP` element in C contains a symbol with CD D. We call a CD **basic**, iff it does not import other CDs.

In contrast to other module systems for Mathematics (see [RK08,RK09b] for an overview) OpenMath does not make make the "imports relation" explicit and in particular does not make any assumptions about the absence of cycles.

Definition 20 (Signature and Property set of a CD). The **signature** of a CD C is the set of symbols it declares in union with the signatures of all CDs imported by C.

Similarly, the **property set** of a CD C is the set of OpenMath objects in `FMP` elements in C (these are called the **local properties** of C) in union with all the axiom sets of all CDs imported by C.

With this, we can directly define the OM theory induced by a CD.

Definition 21 (Theory of a CD). We call the pair (S, P), where S is the signature of C and P is the property set of C the **theory of** C.

In essence, the OM theory of a content dictionary is the union of all symbol declarations and mathematical properties from all theories from which a symbol is used in the CD. There is no problem with the (implicit) imports being cyclic, since their morphisms are the identity and we are constructing the (iterated) union. Note furthermore that OpenMath only supports literal CD names, and we can assume the set of CDs to be finite, therefore, the signature and axiom set of a CD are finite.

Note that in contrast to our definitions from section 3.1, the signature of a CD will already contain the logic, as OpenMath distinguishes logics from regular CDs. Following accepted mathematical practice we assume the logic to be first-order logic (with a choice operator) and a version of axiomatic set theory as a theory of first-order logic — we choose Zermelo Fraenkel set theory with choice [Zer08,Fra22] since this is the best-known one. Note that any other foundation of Mathematics would serve equally well for our purposes. For simplicity of presentation we will assume the existence of two basic CDs for first-order

[6] Arguably the OpenMath standard cannot fix this fully, since it intends to support all mathematical software systems including such that are "semantics-independent" like mathematical editing systems.

logic (declaring connectives, quantifiers, equalities, and choice) and `ZFC` (declaring membership and axioms).

In OpenMath practice, commented mathematical properties seem to assume ZFC as a foundational system, whereas `FMP`s make due with less: they usually only use symbols from the CDS

- `logic1` [CDl04]: a logic in the sense of Definition 14, as it supplies the symbol `true` — which we take to mean $\top$,
- and the symbol `eq` from CD `relation1` [CDr04] — which we take to mean $=$,
- `quant1` [CDq04] that supplies the first-order quantifiers .

Definition 21 allows us to define the meaning of a CD as a class of OM models.

Definition 22 (Model Class and Entailment for CDs). Let C be an OpenMath CD and Θ the theory of C, then the **Model Class of** C is $\mathcal{M}(\Theta)$ and $C \models o$, iff $\Theta \models o$.

We will now turn to the initial semantics again. The restriction to

Definition 23 (Congruence Relation). *Let A be an OM vocabulary over T. A congruence relation on A is a family of equivalence relations on U^A and R^A_n all denoted by $\equiv$ such that (whenever applicable)*

- *if $u \equiv u'$ and $u_i \equiv u'_i$ for $i = 1, \ldots, n$, then*
$$@^A_n(u, (u_1, \ldots, u_n)) \equiv @^A_n(u', (u'_1, \ldots, u'_n)),$$
- *for $l(\sigma) = n$, if $u \equiv u'$, $\mathcal{V}(p) \equiv \mathcal{V}'(p)$ for all $p \in \overline{\sigma}$, and $\mathcal{F} \equiv \mathcal{F}'$, then*
$$\beta^A_K(u, \mathcal{V}, \mathcal{F}) \equiv \beta^A_K(u', \mathcal{V}', \mathcal{F}'),$$
- *if $u \equiv u'$ and $v \equiv v'$, then $\alpha_k(u, v) \equiv \alpha_k(u', v')$,*
- *if $\mathcal{F} \equiv \mathcal{F}'$ and $u_i \equiv u'_i$ for $i = 1, \ldots, n$, then*
$$\mathcal{F}(u, (u_1, \ldots, u_n)) \equiv \mathcal{F}'(u', (u'_1, \ldots, u'_n)).$$

Definition 24 (Quotient Algebra). *Let A be an OM vocabulary over T, and $\equiv$ a congruence relation on A. Then the OM algebra $Q := A/\equiv$ over T is defined by:*

- $U^Q = U^A/\equiv$,
- *R^Q_n is the set of all functions of the form*
$$f : (U^Q)^n \to U^Q,\ f([u_1]_\equiv, \ldots, [u_n]_\equiv) = [F(u_1, \ldots, u_n)]_\equiv$$
for some $[F]_\equiv \in R^A_n/\equiv$,
- *$@^Q_n$, β^Q_K, and α^Q_k are induced by their analogues in A.*

It is straightforward to show that Q is a well-defined OM algebra if A is.

Definition 25 (Induced Congruence). *Let $\Theta = (T, Ax)$ be an L-theory, then we define a congruence relation $\equiv_\Theta$ on $I(L \cup T)$ as follows:*

$$[o]_\alpha \equiv_\Theta [o']_\alpha \qquad \text{iff} \qquad \Theta \models \mathbb{A}(=, o, o') \qquad \text{for } o, o' \in O(L \cup T, ())$$

and

$$\overline{Subs(o)} \equiv_\Theta \overline{Subs(o')} \qquad \text{iff} \qquad \Theta \models \mathbb{A}(=, o, o') \qquad \text{for } o, o' \in O(L \cup T, (x_1, \ldots, x_n)).$$

*We call $\equiv_\Theta$ the **congruence induced by** Θ.*

Lemma 5. *Let $\Theta = (T, Ax)$ be an L-theory, then $\equiv_\Theta$ is indeed a congruence relation.*

Proof. The proof is straightforward.

As a consequence, the following construction is well-defined.

Definition 26 (Initial Model). Let $\Theta = (T, Ax)$ be an L-theory, then $I(\Theta) := I(L \cup T)/\equiv_\Theta$ is called the **initial model for** Θ

The name "initial model" is justified by

Lemma 6. Let Θ be an L-theory. Then for all $o \in O(T, ())$, we have $[\![o]\!]^{I(\Theta)} = [[o]_\alpha]_\equiv$.

Proof. The proof is a straightforward induction on o.

And that finally yields

Theorem 1. *For all $o \in O(\Theta, ())$ we have $I(\Theta)/\equiv_\Theta \models o$, iff $\Theta \models o$.*

Proof. Immediately from Lem. 6 noting that $\Theta \models o$ iff $[o]_\alpha \equiv_\Theta [\top]_\alpha$.

And that yields the main theorem of this section

Corollary 3 (Herbrand Theorem). *Every OM theory has a model that arises as a quotient of the free OM algebra.*

Note that this is exactly the bridging result between the OpenMath objects semantics postulated in the OpenMath2 standard (see Section 1.3) and the traditional foundations of Mathematics (see section 1.2). And with that we can finally define the meaning of OpenMath objects.

Definition 27 (The Meaning of an OpenMath Object). *Let o be an OpenMath object, then we call the union of the theories of the CDs referenced in o the **Theory** of o. If $o \in O(T, ())$ is a ground OM Object, Θ its theory, and M an OM-Model of Θ, then the **meaning of** o **in** M is $[\![o]\!]^M$.*

4 Conclusion

In this paper we have tried to rectify common misunderstandings about the meaning of OpenMath (and thus MathML3) expressions. A central point in the argument can be elucidated by another quote from the referee report mentioned in the introduction — it continued with

> *The [...] "free algebra" semantics is nonsense: it amounts to saying that "the meaning of a term is its syntax". That is* <u>not</u> *what a mathematical semantics is.* anonymous referee for [DK09]

It is our view that the "free" algebra of OpenMath objects forms a kind of initial algebra for "formulae with uninterpreted symbols" which is syntactic in nature as all initial algebras are. Indeed for OpenMath and content MathML expressions that do not contain symbols — and are thus unrestricted by content dictionaries — this is the best meaning we can hope for: OpenMath cannot impose more restrictions than α-equivalence and flattening of attributions without losing coverage. Indeed this is captured by the the algebraic semantics of OpenMath expressions in section 2.

But the meaning of an OpenMath object comes mainly from the mathematical properties in the content dictionaries of its symbols. In section 3 we have been able to show that this can be grafted onto the algebraic semantics by interpreting OpenMath CDs as logical theories over a foundational system like first-order logic with ZFC as an axiomatic set theory.

Note that our semantic analysis has only taken into account symbol names, roles, and mathematical properties. The former two are relevant for the OM vocabularies and the latter for the OM theories that give OpenMath objects their meaning. In particular, we did not look at descriptions (for symbols or whole CDs) or examples. The status of these CD parts is left unspecified, by the OpenMath2 standard, and usage in actual CDs is non-uniform. Symbol descriptions reach from appealing to the folklore — e.g. "*This symbol represents the Boolean value true.* [CDl04] to specific literature references e.g. "*See CRC Standard Mathematical Tables and Formulae, editor: Dan Zwillinger, CRC Press Inc., 1996, (7.7.11) section 7.7.1.*" [CDs04]d. Arguably both forms "mean" something to the human reader, and especially the latter should surely contribute to the theory. The case of examples in CDs is similarly, unclear: if they were uninformative to the human reader, nobody would put them in, but again practice in published CDs is no help. Examples are often statements — and thus in principle mathematical properties — about (mathematical objects constructed by) the symbols they illustrate, and if they are they tend to be valid, but it would be uncautious to assume this to be generally the case. The next version of the OpenMath standard could of course clarify these issues at the cost of making it more heavyweight and thus arguably less useful, we propose to use the OMDoc format [Koh06] that already makes these issues for specifying content dictionaries instead.

A final objection often brought up against the "semantics of OpenMath" is that the standard CDs maintained by the OpenMath society are very weak, and (even with the methods presented here) do not give a clear and unambiguous meaning for K-14 mathematics. Indeed this criticism is formally justified, but misses the main point of the OpenMath philosophy, namely that the set of CDs is open-ended, and that we can build CDs to suit all our communication and representation needs. In particular it is possible (and in fact rather simple) to build a CD `NatArith` for natural numbers and arithmetic by encoding the Peano Axioms and recursive equations for the arithmetical operators in OpenMath objects so that that its theory $\Theta = \Theta(\texttt{NatArith})$ determines the class of Θ-models up to isomorphism. To see this just use the standard proof with our notion of OM models from section 3. If this does not count as clear and unambiguous meaning then what? The OpenMath society (and the W3C Math Working Group for that matter) view the weakness of the standard OpenMath/MathML CD group as a feature and not a bug. These CDs contain fewer mathematical properties to allow them to describe larger model classes. For instance the CD `arith1` [CDa04] (somewhat) corresponds to the class of (Abelian) semigroups. And that is a good thing, since it is intended to capture the informal usage in K-14: in many situations we need the flexibility offered by the OpenMath/MathML CDs so that we do not over-specify the meaning. We would probably not want to scare elementary school children who are struggling with long division with the Peano Axioms or teenagers in high school with the fine differences between Riemann and Lebesque Integration.

We end this treatise on the "meaning of OpenMath and MathML" with the observation that it is possible to specify the meaning of mathematical objects and formulae at many levels of flexibility and rigorousness and extend the invitation to our readers to do just that: to contribute content dictionaries to the community of mathematicians (by way of the OpenMath society).

References

ABC$^+$03. Ron Ausbrooks, Stephen Buswell, David Carlisle, Stéphane Dalmas, Stan Devitt, Angel Diaz, Max Froumentin, Roger Hunter, Patrick Ion, Michael Kohlhase, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical Markup Language (MathML) version 2.0 (second edition). W3C recommendation, World Wide Web Consortium, 2003.

ABC$^+$08. Ron Ausbrooks, Bert Bos, Olga Caprotti, David Carlisle, Giorgi Chavchanidze, Ananth Coorg, Stéphane Dalmas, Stan Devitt, Sam Dooley, Margaret Hinchcliffe, Patrick Ion, Michael Kohlhase, Azzeddine Lazrek, Dennis Leas, Paul Libbrecht, Manolis Mavrikis, Bruce Miller, Robert Miner, Murray Sargent, Kyle Siegrist, Neil Soiffer, Stephen Watt, and Mohamed Zergaoui. Mathematical Markup Language (MathML) version 3.0. W3C working draft of november 17., World Wide Web Consortium, 2008.

Bar91. H. Barendregt. Introduction to Generalized Type Systems. *Journal of Functional Programming*, 1(2):125–154, 1991.

BBK04. Christoph Benzmüller, Chad Brown, and Michal Kohlhase. Higher order semantics and extensionality. *Journal of Symbolic Logic*, 2004.

BCC$^+$04. Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaetano, and Michael Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004.

Ber37. P. Bernays, 1937. Seven papers between 1937 and 1954 in the Journal of Symbolic Logic.

Bou64. N. Bourbaki. Univers. In *Sminaire de Gomtrie Algbrique du Bois Marie - Thorie des topos et cohomologie tale des schmas*, pages 185–217. Springer, 1964.

Can83. G. Cantor. Grundlagen einer allgemeinen Mannigfaltigkeitslehre. Ein mathematisch-philosophischer Versuch in der Lehre des Unendlichen. *Mathematische Annalen*, 1883.

CDa04. arith1. OpenMath Content Dictionary at `http://www.openmath.org/cd/arith1.ocd`, 2004.

CDl04. logic1. OpenMath Content Dictionary at `http://www.openmath.org/cd/logic1.ocd`, 2004.

CDq04. quant1. OpenMath Content Dictionary at `http://www.openmath.org/cd/quant1.ocd`, 2004.

CDr04. relation1. OpenMath Content Dictionary at `http://www.openmath.org/cd/relation1.ocd`, 2004.

CDs04. s_data1. OpenMath Content Dictionary at `http://www.openmath.org/cd/s_data1.ocd`, 2004.

CH88. T. Coquand and G. Huet. The Calculus of Constructions. *Information and Computation*, 76(2/3):95–120, 1988.

Chu40. A. Church. A Formulation of the Simple Theory of Types. *Journal of Symbolic Logic*, 5(1):56–68, 1940.

DK09. James H. Davenport and Michael Kohlhase. Unifying Math Ontologies: A tale of two standards. In *MKM/Calculemus 2009 Proceedings*, LNAI. Springer Verlag, 2009. in Press.

Fra22. A. Fraenkel. The notion of 'definite' and the independence of the axiom of choice. 1922.

Gir71. J. Girard. Une extension de l'interpretation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J. Fenstad, editor, *2nd Scandinavian Logic Symposium*, pages 63–92. North-Holland, 1971.

Göd31. K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931. English title: On Formally Undecidable Propositions Of Principia Mathematica And Related Systems.

Göd40. K. Gödel. The Consistency of Continuum Hypothesis. *Annals of Mathematics Studies*, 3:33–101, 1940.

Hil00. D. Hilbert. Mathematische Probleme. *Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen*, pages 253–297, 1900.

Hil26. D. Hilbert. Über das Unendliche. *Mathematische Annalen*, 95:161–90, 1926.

Koh06. Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.

ML74. P. Martin-Löf. An Intuitionistic Theory of Types: Predicative Part. In *Proceedings of the '73 Logic Colloquium.* North-Holland, 1974.

Rab08. Florian Rabe. *Representing Logics and Logic Translations.* PhD thesis, Jacobs University Bremen, 2008.

Rey74. J. Reynolds. Towards a Theory of Type Structure. In *Paris Colloq. on Programming*, pages 408–425. Springer, 1974.

RK08. Florian Rabe and Michael Kohlhase. An exchange format for modular knowledge. In Piotr Rudnicki and Geoff Sutcliffe, editors, *Knowledge Exchange: Automated Provers and Proof Assistants (KEAPPA)*, November 2008.

RK09a. Florian Rabe and Michael Kohlhase. A better role system for openmath. submitted to the 22nd OpenMath Workshop, 2009.

RK09b. Florian Rabe and Michael Kohlhase. A web-scalable module system for mathematical theories. Manuscript, to be submitted to the Journal of Symbolic Computation, 2009.

Rus01. B. Russll. Recent work in the philosophy of mathematics. *International Monthly*, 1901.

Rus08. B. Russell. Mathematical Logic as Based on the Theory of Types. *American Journal of Mathematics*, 30:222–262, 1908.

Tar38. A. Tarski. Über Unerreichbare Kardinalzahlen. *Fundamenta Mathematicae*, 30:176–183, 1938.

vN25. J. von Neumann. Eine Axiomatisierung der Mengenlehre. *Journal für die reine und angewandte Mathematik*, 154:219–240, 1925.

WR13. A. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1913.

Zer08. E. Zermelo. Untersuchungen ber die Grundlagen der Mengenlehre I. *Mathematische Annalen*, 65:261–281, 1908. English title: Investigations in the foundations of set theory I.