The Freedom to Extend OpenMath and its Utility

James H. Davenport and Paul Libbrecht

Abstract. OpenMath is a standard for representing the *semantics* of mathematical objects. It differs from Presentation MathML in not being directly concerned with the presentation of the object, and from Content MathML 2 in being extensible.

How should these extensions be performed so as to maximize the utility (which includes presentation) of OpenMath? How could publishers have the freedom to extend and let consumers find their way with expressions discovered on the Web? The answer up to now has been, too often, to say "this is not specified" whereas the existing content dictionary mechanism of OpenMath allows it to include formal properties which state mathematical facts that should stay uncontradicted while manipulating the symbols.

The contribution of this paper is to propose methods to exploit the content dictionaries so as to allow an OpenMath-consuming tool to process expressions even if containing symbols it did not know about before. This approach is generalized to allow such newly discovered symbol to be, for example, rendered or input.

1. What is OpenMath?

"OpenMath is an emerging standard for representing mathematical objects with their semantics, allowing them to be exchanged between computer programs,

This paper grew of a discussion at the IMA Workshop "The Evolution of Mathematical Communication in the Age of Digital Libraries" — December 8–9, 2006. Thanks are due to the IMA, and particularly Robert Miner, for organizing this workshop. Section 8 owes a lot to discussion with Prof. Vorobjov. Drs Naylor and Padget also made useful suggestions. Jonathan Stratford kindly proofread a late version.

stored in databases, or published on the worldwide web."[1] In particular, OpenMath is extensible, unlike MathML 2.0[2] [8]. It achieves this by having an extensible collection of Content Dictionaries. "Content Dictionaries (CDs) are used to assign informal and formal semantics to all symbols used in the OpenMath objects. They define the symbols used to represent concepts arising in a particular area of mathematics" [6, section 1.3].

Notation 1. By an *OpenMath CD* we will mean any document conforming to the formal syntax of [6].

The status of an OpenMath content dictionary is one of the following [6, Section 4.2.1]:

- `official`: approved by the *OpenMath* society according to the procedure defined in section 4.5 (of [6]);
- `experimental`: under development, and thus liable to change;
- `private`: used by a private group of *OpenMath* users;
- `obsolete`: an obsolete Content Dictionary kept only for archival purposes[3].

Definition 1.1. A Content Dictionary is said to be *public* if it is accessible from `http://www.openmath.org` and has one of the two status `official` or `obsolete`. Similarly, a symbol is said to be *public* if it is in a public CD.

Note that this definition of *public* refers to the entire symbol, not just the name. Thus `<OMS name="sin" cd="transc1"/>` refers to a public symbol, whereas `<OMS name="sin" cd="transc1" cdbase="http://www.camalsoft.com/G/"/>` is not, since it quotes a `cdbase` other than the official OpenMath one.

An OpenMath object, all of whose symbols are public, has fixed, permanent, semantics. Even if a CD changes status from `official` to `obsolete`, the semantics do not change (though it is quite likely that new software systems will not be able to interpret it, except in the name of compatibility[4]).

The OpenMath standard explicitly envisages that OpenMath applications can declare and negotiate the CDs (or CD groups) that they understand [6, Section 4.4.2]. In the absence of such negotiation, which may well be impossible in a 'cut and paste' scenario, it might seem that the only OpenMath objects which can safely be exchanged are ones all of whose symbols are public (which we can abbreviate to *public* OpenMath objects). This is, indeed, common practice. If every application had to convert from its semantics to those of the public CDs, there would be great inefficiency involved, especially if the aim was 'cut and paste' from one instance of

[1] `http://www.openmath.org/overview/index.html`

[2] After the first version this paper was submitted, a draft [9] of MathML 3.0 was produced, which bases content markup on OpenMath content dictionaries, and thus *is* extensible.

[3] This is the wording of [6]: the present authors would be inclined to write "archival and compatibility purposes".

[4] "Compatibility is the last excuse for not fixing something that you have already admitted to be a bug" [33]. For OpenMath, declaring a CD obsolete and writing a new one with the 'bug' fixed removes even this excuse: see section 7.

an application to another instance of the same application (e.g. from mine to yours, or from today's to tomorrow's, or from version x to version ++x or ...). Equally, two different applications may be "sufficiently similar" that each can understand the other's semantics directly.

Our goal is not to refute the utility of public CDs but to provide freedom and mechanisms of understanding among applications that mostly understand each other. Indeed two applications can use the *semantics* of public CDs without necessarily having to convert every expression into those semantics if they *also* understand each other well enough (as would be normal for different versions of the same software).

2. A Pragmatic Interpretation

Definition 2.1. A Content Dictionary is said to be *semi-public* if it is accessible from `http://www.openmath.org` or from an URI which resolves to a globally accessible URL, and the CD has one of the two status `official` or `obsolete`. Similarly, a symbol is said to be *semi-public* if it is in a semi-public CD.

Thus

```
<OMS name="sin" cd="transc1" cdbase="http://www.camalsoft.com/G/"/>
```

appears to be a semi-public symbol, whereas

```
<OMS name="sin" cd="transc1" cdbase="file://C:/camaljpff/G/"/>
```

is not.

We said above that it *appeared* to be a semi-public symbol. That is because the definition is neither effective (we can try to look the symbol up, but who knows if the failure is transient or permanent) nor time-invariant: `camalsoft` may go bankrupt, or its managers may not comply with the OpenMath rules, and delete symbols or change the semantics of them. Hence the concept that can be effective is that of *apparently semi-public*, as applied to a CD or a symbol. However, an apparently semi-public symbol might not have any discernable semantics.

Definition 2.2. A symbol is said to be *transitively public* if:

1. it is apparently semi-public;
2. its semantics can be deduced in terms of public symbols by (possibly many) applications of Formal Mathematical Properties (FMPs) contained in apparently semi-public CDs.

Again, the definition is not time-invariant, for the same reasons as before. Also, it is not application-independent, since one application might be able to make deductions from FMPs that another could not. *However*, it is the semantics and utility of transitively public symbols that we are concerned with here, since these are the *novel* ones that applications might reasonably encounter, and be expected to 'handle'. This is what, effectively, is implied by the `cdbase` attribute in the `OMS` constructs above and is supported by the OpenMath standard

which defines the unique identifier given a cdbase, cd, and name triple. A tool can request the resource behind the identifier made by the concatenation of these, (`<cdbase>/<cd>#<name>`), either relying on the server negotiating types correctly or adding the necessary `.ocd` termination, and thus fetch the content dictionary with its FMPs, and discover that the symbol is transitively public.

3. An Example — arctan

One hypothetical example would be the following, for the system Derive,[5] whose arctan function differs from the definition in [1]. As pointed out in [10], the two definitions could be related by the FMP in Table 3 on page 20, represented textually below[6]

```
eq (arctan' ( z) , conjugate (arctan (conjugate ( z) ) ) )
```

With this definition, a "sufficiently intelligent" (in fact it need not be that intelligent in this case) system would be able to understand OpenMath emitted from Derive containing Derive arctangents, encoded with the following symbol-reference:

```
<OMS name="arctan" cd="transc1" cdbase="http://www.softwarehouse.com/"/>
```

The designer of the Derive→OpenMath phrasebook is then faced with a set of alternatives.

1. Emit in terms of the public OpenMath symbol from `transc1`. This has the advantage that no Derive CD needs to be written, or, more importantly, maintained and kept available. Assuming[7] that Derive can cancel double conjugation, it means that cutting and pasting from one Derive to another is not significantly more expensive. Some-one who is doing Derive $\overset{\text{OpenMath}}{\longrightarrow}$ MAPLE would find it consistent to see the conjugation appearing. Some-one who is doing Derive $\overset{\text{OpenMath}}{\longrightarrow}$ LaTeX would, however, be distinctly surprised by the results, since the arctan emitted in LaTeX would be (invisibly) one with OpenMath semantics, i.e. complex conjugation might appear in the LaTeX where there was none in the Derive.
2. Emit in terms of the Derive symbol above. This has the disadvantage that the CD[8] needs to be written and kept available. If the recipient is another Derive, it would presumably understand this. If the recipient is a "sufficiently clever" other algebra system conforming to OpenMath's semantics of arctan, the

[5] As already stated in [10], this is not an issue of some algebra systems, such as Maple, being "right" and others, such as Derive, "wrong": merely that Derive has chosen a different set of branch cut behaviours from OpenMath. Provided the definitions are correct, the choice is one of taste, fortified with the occasional dash of Occam's razor.

[6] The text renderings here come from David Carlisle's stylesheet, except that we use ' to denote the *non-standard* arctan.

[7] In this case, this assumption is merely one about Derive, since complex conjugation is trivially an involution. However, many other operations, such as $x \mapsto \frac{1}{x}$ used to transform between tan and cot, are not *trivial* involutions, and algebra systems may well shy away from cancelling them.

[8] And the associated STS [12] file.

FIGURE 1. Definition of an alternative arctan

```
eq (arctan' ( z) , times (divide (one, times ( 2 , i) ) ,
                          ln (divide (plus (one, times (i, z) ) ,
                                      minus (one, times (i,z))))))
```

correct result will be achieved. If it has Derive's semantics, it will either notice this directly, or cancel the double conjugations. If it has different semantics, it will presumably know what to do.

An interesting question is what an OpenMath⟶LaTeX phrasebook with no explicit Derive knowledge will do in this case. It is unlikely to have the semantic processing capability to handle the FMP, though in this case it might. We shall see below plausible actions based on files closely related to the content-dictionaries.

3. Ignore the problem, and emit `<OMS name="arctan" cd="transc1"/>`. Alas, this would be a very human reaction. Such a phrasebook would (if it met the other criteria) be entitled to describe itself as OpenMath-compliant, but it would certainly not meet the goal [6, Chapter 5] that "It is expected that the application's phrasebooks for the supported Content Dictionaries will be constructed such that the properties of the symbol expressed in the Content Dictionary are respected as far as possible for the given application domain".
4. Refuse to emit arctans, on the grounds that Derive's is different from OpenMath's. In view of the plausible solutions in the first two choices, this seems unnecessarily "dog-in-the-manger".

We should observe that the mathematically equivalent FMP (see Table 4 on page 21 for the full OpenMath)

```
eq (arctan ( z) , conjugate (arctan' (conjugate ( z) ) ) )
```

is less useful, as it expresses the 'known' `<OMS name="arctan" cd="transc1"/>` in terms of the 'unknown', rather than the other way round, and therefore requires more logical power to use. In particular, the interpreting phrasebook would need to know that the inverse of conjugation is itself conjugation.

Note also that there is no need to define Derive's arctan in terms of the OpenMath one: we could define it directly (see Figure 1 or table 5) in terms of log, as OpenMath's arctan is in `transc1`.

4. Resolution of Expressions with Unknown Symbols

The example above reveals best practice in terms of publishing a content dictionary and receiving an expression with a symbol yet unknown. Let us consider a tool that knows mostly only about official symbols and that this tool is fed with an OpenMath expression containing a transitively public symbol such as software-house's `arctan`, identified by its `name`, `cd`, and `cdbase` attributes.

The OpenMath specification indicates that, then, the canonical URI of this symbol is the concatenation of the value of the `cdbase` attribute, followed by '/', followed by the value of the `cd` attribute, followed by '#' followed by the value of the `name` attribute.

The tool can attempt to fetch the content-dictionary file behind this identifier, that is, it can request the URI `<cdbase>/<cd>` indicating the expectation of the content-dictionary mime-type[9], and, if that fails, attempt `<cdbase>/<cd>.ocd`. If this is successful it can, then, use the FMPs:

- if wishing to render the expression, it may either choose a generic approach (e.g. the prefix notation), or use the `FMP` before rendering with official symbols
- within an evaluation task, such as a plotting or computation task, it may try to directly apply the formal properties
- other tasks may take advantage of the abstract nature of formal properties, an example includes canonicalization of expressions, e.g., for search purposes as in [34] or [25]
- even results of computations may be rendered back using this symbol, again by the applications of formal properties

As we see above, this leaves great latitude to tools' exploitation of `FMP`s which might or not give results a user expects. The *sudden* appearance of a complex conjugation within the rendering of arguments of an arctan function is an example of surprising effect. Moreover, the ability to put the symbol to good use may depend heavily on the computational capabilities of the receiving tool.

The examples above are computationally simple. A more convoluted usage of FMP, is that of the symbol `Stirling1` of the content-dictionary `combinat1`[10]: in order for a recipient to evaluate `Stirling1(n,m)`, one has to apply the FMP of `Stirling1`, followed by the FMP of `Stirling2`, followed by the FMP of `binomial`; similarly, in order to evaluate `Fibonacci(k)`, one needs to apply the FMP of this symbol `k` times, if `k` is known...

Thus we see that it is impossible to dictate *a priori* the complexity of the usage of an FMP and we should leave it to individual tools to succeed in using FMPs or not.

5. Another Example

Let us imagine a theorem prover specialized for statements over the natural numbers: let us call it Euclid. Euclid's natural domain of reasoning is the positive integers $1, 2, \ldots$, which it refers to as $\mathbf{N}$. How should Euclid export results such as "if something times a equals that same something times b, then $a = b$", i.e.

$$\forall a, b, c \in \mathbf{N} \quad a{\cdot}c = b{\cdot}c \Rightarrow a = b \ ? \tag{1}$$

[9] which defaults to `application/xml+openmath-cd` but still needs standardization.
[10] See `http://www.openmath.org/cd/combinat1.xhtml#Stirling1`.

Again, the designer of the Euclid→OpenMath phrasebook is facing various options.

1. Emit in terms of the OpenMath **N** public symbol, i.e. encode Euclid's **N** as:

```
<OMA>
  <OMS name="setdiff" cd="set1"/>
  <OMS name="N" cd="setname1"/>
  <OMA>
    <OMS name="set" cd="set1"/>
    <OMS name="zero" cd="alg1"/>
  </OMA>
</OMA>
```

   which is commonly denoted by $\mathbf{N}\backslash\{0\}$. This is certainly accurate, but would cause some grief on re-importing into Euclid, since:
   - **N** (in the OpenMath sense) has no direct equivalent in Euclid, but has to be encoded as $\mathbf{N} \cup \{0\}$;
   - while expecting an algebra system to cancel double conjugations is reasonable, expecting a proof system to simplify $(\mathbf{N}\backslash\{0\})\cup\{0\}$ is expecting rather more.
2. Emit in Euclid's own CD, e.g. with a definition as below (or Table 6 on page 23).

   ```
   eq (P, setdiff (N, set (zero) ) )
   ```

   This has advantages as well as disadvantages.
   - Clearly it requires the CD to be written and maintained.
   - An OpenMath→LaTeX converter would probably render this as P. This might look well, but could be confused with

     ```
     <OMS name="P" cd="setname1"/>
     ```

     which is the set of primes[11], normally rendered as $\mathcal{P}$. A configurable OpenMath→LaTeX converter[12] would be able to get this right, and print **P** if properly configured for the case.
3. Ignore the difficulty. This is clearly sub-human, rather than merely human, since a theorem-prover that emits incorrect statements could well be argued to be worse than useless.

We return to this issue in section 7.

6. OpenMath and Notation

What use is OpenMath if one can't "see"[13] the results? Probably not much. How does one do the rendering?

---

[11] This is another example of the fact that an OpenMath symbol is the name *and* the CD.
[12] Such as the Notation Selection Tool [28, 29].
[13] "See" and "rendered" are used as shorthand for "convert into a presentation", which may be displayed in various means, e.g. audio [30].

Nowadays, many feature-rich converters from OpenMath to many presentation languages exist; they output to MathML-presentation, TEX, or HTML; they are extensible, for most, at development time. However, only a few such converters allow naïve users to choose among notations as in [29], several make it possible to enrich the notations by an appropriate input in a language close to their programming language, for example the computer-algebra-systems. Only [27] and [24] propose a declarative format to encode the notations.

6.1. Notational Diversity

The MathML content specification, and most books, provide *default notations* information for each symbol. However, this begs the question: what is "the notation" [13]. A simple example is that of half-open intervals: the "anglo-saxon" $(0, 1]$ and the "french" $]0, 1]$. More subtly, there is the "anglo-saxon" use of Arcsin to denote a multi-valued function and arcsin to denote the corresponding[14] one-valued function, compared with the "french" notation which is the converse. It should be noted that, in this case, the OpenMath notation is even-handed: one is

```
<OMS name="arcsin" cd="transc1"/>
```

the other is

```
<OMS name="arcsin" cd="transc3"/>
```

and in both the "anglo-saxon" and "french" cases, one (or one's renderer) has to decide which to capitalize.

To avoid the charge of (anti)gallicanism being levied against the authors, let us also point out that there are differences due to subject: $\sqrt{-1}$ is i everywhere except in electrical engineering, where it is j, and so on. Ambiguity can work both ways: the first author had great difficulty with the notation when applying Gröbner bases, where $[X]$ denotes a polynomial ring in the variable X, to enzyme kinetics, where $[X]$ denotes the concentration of substance X [5].

Thus we see that notations are the result of the fruitful imaginations of each author and that, just as OpenMath needs to support an extensible set of symbols, it needs to support an extensible set of notations.

Moreover, we see that it is impossible for an OpenMath object to know, in a context-free way, how it should be rendered. The best one could hope for is that, associated with an OpenMath CD, there could be a "default rendering" file, which would give *a* rendering for objects using this system, probably by translation into Presentation MathML as in David Carlisle's excellent style sheets [7]. This would have the advantage of allowing technologies such as those described in [20, 30] to process it.

[14] But almost always with the branch cuts locally implicit, and often never stated at all, or changing silently from one printing to the next, as in [1].

6.2. Notation Documents

From what we can see above, in particular in the `arctan` example, there is a requirement for tools discovering new symbols on the web to be able to render them. Without proper rendering instruction, our tool, receiving softwarehouse's `arctan(x)` could either use the default prefix-notation or could use the formal properties, thus rendering it into arctan $\bar{x}$. It should, thus, be possible to make available to discovering software, hints for the rendering of content-expressions. That is one of the goals of notation-documents: it should allow the easy discovery of *notations* just as OpenMath content-dictionary files allow the discovery of new symbols, and this anywhere on the web.

Another requirement of notation-documents made evident in the Euclid example is the strong need for the management of *contexts* to attach the set of accepted notations to. These contexts are, typically, corpora of documents known to a community. The ability to associate a notation-document with a context is, thus, very important. Within the notation-documents, potentially, the need to identify contexts, by name, by well-known values (such as language) is also desirable.

The `.ntn` documents approaches (see [22], or [19]) propose an answer to these requirements: they specify an XML syntax which describes notations in a form as simple as a formal mathematical property: a simple pair of an OpenMath expression (the *prototype*) and its corresponding MathML-p expression (the *rendering*).

An extract of a notation document is provided in the example below, it specifies the rendering of the open-interval (this is extracted from the MathML-3 working draft, chapter 8[15]):

```
<mcd:notation>                    <mcd:rendering>
  <mcd:prototype>                   <math>
    <OMA>                             <mrow><mo>]</mo>
      <OMS cd="interval1"             <mcd:render name="a" precedence="10"/>
         name="interval-oo"/>         <mo>,</mo>
      <mcd:expr name="a"/>            <mcd:render name="b" precedence="10"/>
      <mcd:expr name="b"/>            <mo>[</mo>
    </OMA>                            </mrow>
  </mcd:prototype>              </math></mcd:rendering>
                              </mcd:notation>
```

Such a notation is applied when rendering `interval-oo(n, plus(n,1))`, by the output of the corresponding MathML-presentation expression with `render` elements replaced by the rendering of `n` and of `plus(n,1)`; numeric precedence attributes make it possible to elide unnecessary brackets yielding probably the final rendering $]n, n+1[$.

Notation-documents are expected to be companions of content-dictionaries, typically published in a `.ntn` file where the content-dictionary file is published in the `.ocd` file. But they could also happen to be added in some other places, for example, to indicate the notations for a specific document.

---

[15] Published at `http://www.w3.org/TR/2007/WD-MathML3-20071214/chapter8.html`

Experience developed in the LeActiveMath project has shown that packaging all the notations with notation-documents is a scalable solution. This project has assembled about 500 classical notations for calculus using the mechanism of [24], the server assembles them in an XSLT stylesheet which is then used to render all OpenMath content with very acceptable performance.[16]

Continuing the fetching strategy described in §4, a tool discovering a new symbol would fetch the content-dictionary file at `<cdbase>/<cd>` (or `<cdbase>/<cd>.ocd`), it would also fetch the corresponding notation-document at `<cdbase>/<cd>` (or `<cdbase>/<cd>.ntn`). Doing so and trying to render, or to make an input-button for, the newly discovered symbol, it can choose to:

- use its own methods of symbol rendering, ignoring the notation-document. This would be very wise if it knew well that it owns the best knowledge for rendering complex expressions (often the case with basic symbols such as `plus` which can benefit from substantial case-specific tuning)
- search through the notations that correspond to the expression to render and choose the best notation, helped, among others, by the annotations of context inside the notation-document (for example indicating to use `arctg` instead of `arctan` if identifying the French language as dominant in the delivery context).
- use a generic rendering mechanism, avoiding reading any rendering information: this works well in English for softwarehouse's arctan but much less well with the Euclid case which should in some cases be noted **N** or be noted **P**.

6.3. Security Concerns of the Download Notations

Downloading notations from the web raises the trustability concern of classical web-browsing: indeed, in the situation of the transfer of a user of an OpenMath expression with new symbols into his desktop application, one would expect the application to fetch the content-dictionaries and notations on the web and use them without interaction with the user. Note that only a weak relationship binds the author of the OpenMath fragment being transferred and the content-dictionary author (that of *meaning the same* by using the same symbol). The receiving application could be thus fetching from completely untrusted sources, including from web-servers that have been filled with viral worms.

If the notations are expressed as program fragments an environment should be provided in order for these fragments to act:

- without damage to the user's environment
- without impact on other notations
- in a compatible way with the existing rendering environment, e.g. only using the *known shared routines*.

[16] A list of prototypes of all notations can be seen at `http://demo.activemath.org/ActiveMath2/tools/symbolpresentation.cmd?collections=openmath-cds&collections=LeAM_calculus&noApplet=true`.

```
<mcd:notation>
  <mcd:prototype>
    <OMS cd="Euclid" name="P"/>
  </mcd:prototype>
  <mcd:rendering>
    <mtext>|N</mtext>
  </mcd:rendering>
</mcd:notation>
```

```
<mcd:notation xml:lang="en">
  <mcd:prototype>
    <OMS cd="Euclid" name="P"/>
  </mcd:prototype>
  <mcd:rendering>
    <msup><mtext>|N</mtext>
    <mtext>*</mtext></msup>
  </mcd:rendering>
</mcd:notation>
```

TABLE 1. The notation documents for Euclid's P in the default notation (left) and in the English notation (right), supposing that they have opted for the alternative 2, i.e. to emit their own CD.

Sandboxing mechanisms should thus be accommodated such as, at least, the disablement of extension functions within an XSLT environment.

In comparison to program fragments which have been proposed as solutions, the notations in files of a declarative format such as [22] or [19], have a much more predictable and manageable impact.

6.4. Respecting User's Notations on the Web

Another important aspect of mathematical notations, as we have seen in the Euclid example, is the diversity of notations which is strongly rooted in the cultural diversity of each user's learned mathematics. If one allows new notations to be declared and fetched it should also be possible to support particular notations within particular worlds. For example, notations are often specialized within the context of a text-book. Such notations, should, then, ideally, also be portable to a computing engine that would work for this text-book.

For this to be realized, on the one hand, notations should be writable for specialized contexts, for example, for each natural language (where names of functions may differ). Moreover, users or authors may want to override existing notations. The method proposed in [22] is to define the rendering engine by a sequence of notation documents, the last being of higher priority; [17] proposes to attach notations to documents which can be imported similarly to the textual declaration of each notation document within a mathematical text.

An example usage would be students wishing to use the theorems of the Euclid system in a German context: they will want to use the German notation, whereas students in other languages will want to use the other. Supposing Euclid would export its content dictionary with the German notation, a British student using it would choose the notation of his course: his lecturer, or the student himself, would choose to prioritize the notations for the British context. A simple way could be as depicted in table 1 which presents an extract of the notation document in one and the other document.

6.5. Bundling Notations

The example above has described two forms of bundling of the notations: with or without the content-dictionary and this is the bundling approach that the authors suggest, namely, to provide `.ntn` files *close to* every provision of an apparently semi-public content-dictionary. Doing this guarantees that the newly encountered symbols can be displayed.

However, for the British notation of Euclid's set of integer numbers to be writable, notations need to be fetched from a different place, one close to the *user* rather than close to the content dictionary. This requires a tuning of the engine that fetches notations which can apply the simple approach of overriding notations with notations from *closer* files. Such tunings are means for adaptivity to the user's needs, they are specific to individual rendering applications deployed for each user and are, thus, outside the scope of this paper: we merely point out their utility.

We see the need for standalone notation documents or *virtually standalone* documents as in [17] which bind notations along with the theory-import mechanism of OMDoc where the notations are embedded or referenced as well as the need for documents closely related to content dictionary files.

7. Is Even-handedness Possible?

So far we have tried to be even-handed between various notations: OpenMath makes no choice between $(0, 1]$ and $]0, 1]$, nor says whether the mathematical Arcsin is a single-valued or multi-valued function, i.e. whether it corresponds to the `arcsin` from `transc1` or `transc3`. Even in the case of the branch cuts for arctan, where OpenMath has chosen one definition, it is possible to state the other definition, and do so on an even footing with OpenMath's own definition in `transc1`. Indeed it is possible that, as a result of the great Branch Cut Riots of 2036[17], `transc1` is declared `obsolete`, `transc4` is promulgated with an FMP for arctan as in Figure 1 on page 5, and the authors of the softwarehouse CD change the FMP for arctan to declare this equal to the `transc4` one (Table 7 on page 23) and probably also mark their CD as `obsolete`. None of this would change the semantics of any OpenMath object.

However, the problem raised in section 5 is not so easily resolved: the question of whether $\mathbf{N}$ contains zero can, and indeed has [14], generate much debate. Many books, especially in German, suppose that $\mathbf{N}$ does *not* contain zero, e.g. the following.

> Natürliche Zahlen sind die Zahlen, mit denen wir zählen: 1, 2, 3, 4, 5,.... Auf der Zahlengeraden bilden sie eine Abfolge von Punkten im Abstand 1, von 1 aus nach rechts gehend. Die Menge aller

[17] Caused by the requirement to move the branch cut in Network Time Protocol [26] and associated data formats. Rioters marched under the slogan "give us our two thousand one hundred and forty seven million, four hundred and eighty three thousand, six hundred and forty eight seconds back".

> natürlichen Zahlen wird mit $\mathbf{N}$ bezeichnet. Weiters verwenden wir die Bezeichnung $\mathbf{N}_0 = \{0\} \cup \mathbf{N}$ für die natürlichen Zahlen zusammen mit der Zahl 0. [2, N]

Other sources are less definitive.

> Die natürlichen Zahlen sind die beim Zählen verwendeten Zahlen 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, usw. Oft wird auch die 0 (Null) zu den natürlichen Zahlen gerechnet. [3, Natürliche Zahl].

Indeed, the question is apparently as context-dependent as the rendering of $\sqrt{-1}$, but the impact of getting it wrong is much more misleading.

> Even German school books differ here. It depends on whom you ask. If you ask someone from number theory, he'd usually say that N is without 0. But if you ask someone from set theory, he'd say that N is with 0. It's just what is more convenient (i.e. shorter) for their usual work. [15]

It is clear that we have two different concepts, and several notations, as shown in Table 2.

TABLE 2. Natürliche Zahl

Concept	English	German (number)	German (set)	OpenMath
$0, 1, 2 \ldots$	$\mathbf{N}$	$\mathbf{N}_0$	$\mathbf{N}$	`name="N" cd="setname1"`
$1, 2, 3 \ldots$	$\mathbf{N}^+$ or $\mathbf{N}^*$	$\mathbf{N}$	?	`??`

What should replace `??`. Following our earlier policies, that different concepts (like one-valued/multi-valued arcsin) have different OpenMath, it clearly has to be a new symbol. With hindsight, the German number-theory notation might have been the best to inspire OpenMath, but we cannot change the semantics of `<OMS name="N" cd="setname1"/>`. We could introduce a new `N` in a different CD, and declare `setname1` obsolete, but that would probably be worse than the Branch Cut Riots.

Hence we need another symbol. This could be in `setname1`, or in some other CD. If in `setname1`, it would need another name: if in another CD, it could also be called N, but this would probably cause more chaos. So, let us propose that we add

```
<OMS name="Nstar" cd="setname1"/>
```

to OpenMath. We then have a choice: we can define it in terms of the standard $\mathbf{N}$, as we suggested in Table 4, or we can define it in a free-standing way, by saying that it is 1 and its successors: formally (or Table 8)

```
forall [n].(implies (in ( n, Nstar) ,
   or (eq ( n, one) , in (minus ( n, one) , Nstar) ) ) )
```

(it being assumed here, as in the case of the existing definition of $\mathbf{N}$, that this definition is minimal, i.e. Peano's axioms).

Provided we have *at least* the second definition (having both is not excluded), we are being as even-handed as possible: both concepts exist in OpenMath, as in the case of single-valued/multi-valued arcsin. Admittedly, the *default* rendering *might* be of $0\ldots$ as $\mathbf{N}$, and $1\ldots$ as $\mathbf{N}star$ or $\mathbf{N}^*$, but this is merely another reason for renderers to be configurable.

8. Semantics Drives Notation?

So far, this paper has argued that semantics is all that matters, and that notation should follow. This is essentially the OpenMath premise (and the authors'). But life has a habit of not being so simple: take 'O'. Every student is taught that $O(f(n))$ is really a set, and that when we write "$g(n) = O(f(n))$", we really mean "$g(n) \in O(f(n))$". Almost all[18] textbooks then use '=', having apparently placated the god of Bourbaki[19]. However, actual uses of O as a set are rare: the authors have never[20] seen "$O(f) \cap O(g)$", and, while a textbook might[21] write "$O(n^2) \subset O(n^3)$", this would only be for pedagogy of the O-notation. So 'O' abuses notation, but OpenMath is, or ought to be, of sterner stuff. It certainly *would* be an abuse of `<OMS name="eq" cd="relation1"/>` to use it here, as the relation it implies is none of reflexive, symmetric or transitive[22].

The set-theoretic view is the one taken by OpenMath CD[23] `asymp1`, except that only limiting behaviour at $+\infty$ is considered[24], and there is some type confusion in it: it claims to represent these as sets of functions $\mathbf{R} \to \mathbf{R}$, but in fact the expressions are assuming $\mathbf{N} \to \mathbf{R}$.

Hence it is possible to write $\lambda n.n^2 \in O(n^3)$ in OpenMath. This poses two problems for renderers:

a) how to kill the λ;
b) how to print '=' rather than '$\in$'.

The first problem is common across much of mathematics: remark that $\lambda m.m^2 \in O(n^3)$ is equally valid, but one cannot say $m^2 = O(n^3)$.

The second problem could be solved in several ways.

1. By resolutely using $\in$, as [21].

[18] [21] is an honourable exception.
[19] "the abuses of language without which any mathematical text threatens to become pedantic and even unreadable".
[20] Not even in the one context where it *would* be useful: $\Theta(f) = O(f) \cap \Omega(f)$, which is stated in words as [11, Theorem 3.1].
[21] [11, p. 41] write $\Theta(n) \subset O(n)$.
[22] Curiously enough, the FMPs currently only state transitivity: this probably ought to be fixed.
[23] Currently `experimental`.
[24] The CD author presumably considered that the level of abstraction needed for a more general definition was unwarranted. The current authors would agree, especially as the context of O is generally only implicit in the wider context of the paper.

2. By attributing to each appropriate use of `<OMS name="in" cd="set1"/>` its print representation.
3. By fixing the rendering of `<OMS name="in" cd="set1"/>` to print it as '=', either:
   (a) for all symbols in `asymp1` (thus getting it "wrong" for subsequently invented[25] symbols such as
   `<OMS name="soft0" cd="asymp2"/>`
   which would lie in different CDs);
   (b) or for all usages of the (STS or other) type "function in set", but this would print $\sin = \mathbf{R}^{\mathbf{R}}$ rather than $\sin \in \mathbf{R}^{\mathbf{R}}$.
4. (the authors' favourite) By adding a symbol[26]
   `<OMS name="Landauin" cd="asymp1"/>`,
   which would, by default, print as '=', but have the semantics of '$\in$'.

   How is this last to be achieved?
   One possibility would be to say that it is the same as '$\in$':

```
<FMP>
<OMOBJ cdbase="http://www.openmath.org/cd">
  <OMA>
    <OMS cd = "relation1" name="eq"/>
    <OMS cd = "set1" name="in"/>
    <OMS cd = "asymp1" name="Landauin"/>
  </OMA>
</OMOBJ>
</FMP>
```

but this runs the risk of saying that any '$\in$' can become `Landauin`. A better way might be (Table 9)

```
implies (Landauin ( A, B) , in ( A, B) )
```

or to say that `Landauin` is a *special-case* of $\in$: hypothetically as

```
<FMP>
<OMOBJ cdbase="http://www.openmath.org/cd">
  <OMA>
    <OMS cd = "generalizations" name="special-case"/>
      <OMS cd = "asymp1" name="Landauin"/>
      <OMS cd = "set1" name="in"/>
    </OMA>
  </OMA>
</OMOBJ>
</FMP>
```

[25] [32] apparently introduced this: 'where the "soft O" $\tilde{O}$ indicates an implicit factor of $(\log n)^{O(1)}$'.
[26] It might be more appropriate to call it `Bachmannin`, since [4] is apparently the source of O. [18]

using a symbol that is yet to be declared to denote the *special-case* relation. This would have at least the advantage that applications which are not too versed into calculus could still interprete such a symbol as `Landauin`.

9. Conclusions

OpenMath *can* represent a variety of concepts, not just those "chosen by the designers". Alternative choices of branch cuts, single-valued/multi-valued functions, starting point for the natural numbers etc. are all supportable. Whether these are rendered in a manner appropriate to the user clearly depends on the usage and delivery platform, which means that OpenMath renderers *need* to be configurable, and at a variety of levels.

Even the Bourbaki school believe that notation exists to be abused, as well as used: OpenMath exists purely to be used, and does not exist to be abused. However, in some cases such as 'O', it may need to make slight adjustments to permit conventional notation, such as inserting symbols like `<OMS cd = "asymp1" name="Landauin"/>`, which are mathematically redundant.

This article has taken the reader through a few symbols and notations which show well the flourishing imagination of mathematics authors. It has proposed methods to allow authors defining new symbols and new notations to publish them, and recipients to face them.

It promotes an approach of reasoning about properties of semantic objects declared in documents published on the web. This is similar to the approach of the web-ontology-language OWL [16] for which very many ontology files are available around the web. OWL reasoning falls into three categories: OWL-Lite, OWL-DL, and OWL-Full, but the formal-properties of OpenMath symbols published on the web are not specifiable under such constraints. Exploring the relationship between OWL-reasoning and OpenMath would be a useful piece of further research.

We expect that a managed extensibility as we describe in this paper will empower collaborations between encyclopedia makers and desktop computing systems makers.

- encyclopedia makers, such as, for example, those of the KnotInfo table of knot invariants Web-site[27], have a constant need to update the language of mathematical expressions used in their web-site, and this need may appear in a way that is far from being under their control (for example by an upgrade of the WebMathematica back-end).
- desktop computing systems makers need to be able to perform user-expected operations on any mathematical objects they receive: reacting to new symbols in a way that does not make them *just another unknown* is required. We expect the ubiquity of the world wide web to push them in this direction.

[27] `http://www.indiana.edu/~knotinfo/`

The current choice of implementation is dictated by practice: simple linear syntax for the output of the KnotInfo encyclopedia as well as the input of the computing engine. In some cases, though, it is not clear if that linear syntax can be shared at all, e.g., if the computing system is to compute with the full symmetry group of the knot called $12n\_0276$ which happens to be called $D6$.

In [23] Steve Linton exposed issues of *putting a group into the wire*, that is, of the OpenMath serialization of the information about a group coming for example from GAP: this information can become quickly very big and it is never clear which facet(s) of the group is wished (e.g. a compact *generators and relations* or a multiplication table!). The methods of exposing symbols and formal properties on the web described in this article could be taken further to provide an elegant and compact way to export the information known about a group. Doing this may require some element of laziness ("I know the character table of this group, but I won't tell you unless you explicitly ask for it"), and doing this in an elegant manner is again a question for future research.

9.1. Detailed Suggestions for OpenMath

1. Add `<OMS cd = "asymp1" name="Landauin"/>`.
2. Add reflexive and symmetric properties to equality `<OMS cd = "relation1" name="eq"/>`.
3. Add `<OMS name="Nstar" cd="setname1"/>`, possibly to `setname1` or possibly to another CD.
4. Add a standard means of giving printing attributes (as required in point 2 on page 15).
5. Add a standard means of carrying notations in documents as easily fetched as content-dictionary files by standardizing approaches such as [24], [19], [22].
6. Request that applications' and phrasebooks' developers document the set of content-dictionaries they support so as to help intermediate services to use formal properties, to receive expressions from the world.
7. create a content-dictionary that expresses such relationships as specialization (see section 8). This *might* form part of a wider project to explore how OWL-like reasoning might interact with OpenMath.

References

[1] M. Abramowitz and I. Stegun. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. *US Government Printing Office*, 1964.

[2] Anonymous. Natürliche Zahlen. `http://www.mathe-online.at/mathint/lexikon/n.html`, 2006.

[3] Anonymous. Wikipedia, Deutsch. `http://de.wikipedia.org`, 2007.

[4] P. Bachmann. Die analytische Zahlentheorie. *Teubner*, 1894.

[5] J.P. Bennett, J.H. Davenport, M.C. Dewar, D.L. Fisher, M. Grinfeld, and H. Sauro. Computer Algebra Approaches to Enzyme Kinetics. in Algebraic Computing in Control, Springer Lecture notes in Control and Information Sciences 165, 1991 G. Jacob and F. Lamnabhi-Lagarrigue (eds)

[6] S. Buswell, O. Caprotti, D.P. Carlisle, M.C. Dewar, M. Gaëtano, and M. Kohlhase. The OpenMath Standard 2.0. `http://www.openmath.org`, 2004.

[7] D.P. Carlisle. Openmath, MathML and XSLT. *ACM SIGSAM Bulletin 2*, 34:6–11, 2000.

[8] World-Wide Web Consortium. Mathematical Markup Language (MathML) Version 2.0 (Second Edition). *W3C Recommendation 21 October 2003*, 2003. `http://www.w3.org/TR/MathML2/`.

[9] World-Wide Web Consortium. Mathematical Markup Language (MathML) Version 3.0. *W3C Working Draft*, 2007. `http://www.w3.org/TR/2007/WD-MathML3-20070427`.

[10] R.M. Corless, J.H. Davenport, D.J. Jeffrey, and S.M. Watt. According to Abramowitz and Stegun. *SIGSAM Bulletin 2*, 34:58–65, 2000.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to Algorithms, 2nd. ed.. *M.I.T. Press*, 2001.

[12] J.H. Davenport. A Small OpenMath Type System. *ACM SIGSAM Bulletin 2*, 34:16–21, 2000.

[13] J.H. Davenport. Nauseating Notation. `http://staff.bath.ac.uk/masjhd/Drafts/Notation.pdf`, 2007.

[14] E.W. Dijkstra. Why numbering should start at zero. `http://www.cs.utexas.edu/users/EWD/transcriptions/EWD08xx/EWD831.html`, 1982.

[15] C. Gross. Re: Utility of OpenMath. E-mail `64191.89.49.160.232.1172849601.squirrel@webmail.uni-augsburg.de`, 2007.

[16] Deborah L. McGuinness and Frank van Harmelen eds., OWL Web Ontology Language Overview W3C Recommendation 10 Feb 2004

[17] M. Kohlhase, C. Müller, N. Müller. Documents with flexible Notation Contexts as Interfaces to Mathematical Knowledge Proceedings of MathUI 2007 `http://www.activemath.org/workshops/MathUI/07/`, 2007.

[18] D.E. Knuth. Big Omicron and big Omega and big Theta. *ACM SIGACT News 2*, 8:18–24, 1974.

[19] C. Lange. OM Presentation with Flexible Elisions OpenMath Workshop, Linz, 2007 `http://jem-thematic.net/node/164`.

[20] A. Lazrek. Multilingual Mathematical e-Document Processing. `http://www.ima.umn.edu/2006-2007/SW12.8-9.06/activities/Lazrek-Azzeddine/MathArabIMAe.pdf`, 2006.

[21] A. Levitin. Introduction to the design and analysis of algorithms. *Pearson Addison-Wesley*, 2007.

[22] P. Libbrecht, Content Dictionary Notations, OpenMath Workshop, Linz, 2007 `http://jem-thematic.net/node/328`

[23] S. Linton. Interfacing with GAP. Talk presented in the Mathematics on the Semantic Web Workshop, Eindhoven, May 12-14 2003 (http://www.openmath.org/meetings/eindhoven2003/index.html), 2003.

[24] S. Manzoor, P. Libbrecht, C. Ullrich, and E. Melis. Authoring Presentation for OPENMATH. In M. Kohlhase, editor, *Proceedings MKM 2005*, pages 33–48, 2005.

[25] R. Miner and R. Munavalli, An Approach to Mathematical Search Through Query Formulation and Data Normalization Proceedings of the Mathematical Knowledge Management Conference, Linz, 2007. See http://www.springerlink.com/content/f586745p31j265w4.

[26] D.L. Mills. Network Time Protocol, Version 3. http://rfc.net/rfc1305.html, 1992.

[27] W. Naylor, S. Watt. Meta Stylesheets for the Conversion of Mathematical Documents into Multiple Forms Proceedings of Mathematical Knowledge Management Conference 2001. Available at http://www.cs.bath.ac.uk/~wn/Papers/MetaMML/metaMML+/index.html.

[28] E. Smirnova and S.M. Watt. Interfaces for Mathematical Communication. http://www.ima.umn.edu/2006-2007/SW12.8-9.06/activities/Smirnova-Elena/SmirnovaWatt.pdf, 2006.

[29] E. Smirnova and S.M. Watt. Notation Selection in Mathematical Computing Environments. In *Proceedings Transgressive Computing 2006*, pages 339–355, 2006.

[30] N. Soiffer. Accessible Mathematics. http://www.ima.umn.edu/2006-2007/SW12.8-9.06/activities/Soiffer-Neil/index.htm, 2006.

[31] Various. LeActiveMath. http://www.activemath.org, 2007.

[32] J. von zur Gathen. Irreducibility of multivariate polynomials. *J. Computer Syst. Sci.*, 31:225–264, 1985.

[33] D.J. Wheeler. Private Communication to J.H. Davenport. *June* 1982.

[34] M.E. Altamimi and Abdou S. Youssef A More Canonical Form of Content MathML to Facilitate Math Search. Extreme Markup Languages 2007 http://www.idealliance.org/papers/extreme/proceedings/html/2007/Altamimi01/EML2007Altamimi01.xml

Table 3. An FMP for arctan

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
    <OMA>
      <OMS name="eq" cd="relation1"/>
      <OMA>
        <OMS name="arctan" cd="transc1" cdbase="http://www.softwarehouse.com/Derive/"/>
        <OMV name="z"/>
      </OMA>
      <OMA>
        <OMS name="conjugate" cd="complex1"/>
        <OMA>
          <OMS name="arctan" cd="transc1"/>
          <OMA>
            <OMS name="conjugate" cd="complex1"/>
            <OMV name="z"/>
          </OMA>
        </OMA>
      </OMA>
    </OMA>
  </OMOBJ>
```

James H. Davenport
Department of Computer Science, University of Bath, Bath BA2 7AY England
e-mail: J.H.Davenport@bath.ac.uk,
WWW home page: http://staff.bath.ac.uk/masjhd

Paul Libbrecht
DFKI GmbH, Saarbrücken, Germany
e-mail: paul@activemath.org.org
WWW Home page: http://www.activemath.org/~paul

TABLE 4. An alternative, less useful, FMP for arctan

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
    <OMA>
      <OMS name="eq" cd="relation1"/>
      <OMA>
        <OMS name="arctan" cd="transc1"/>
        <OMV name="z"/>
      </OMA>
      <OMA>
        <OMS name="conjugate" cd="complex1"/>
        <OMA>
          <OMS name="arctan" cd="transc1" cdbase="http://www.softwarehouse.com/Derive"/>
          <OMA>
            <OMS name="conjugate" cd="complex1"/>
            <OMV name="z"/>
          </OMA>
        </OMA>
      </OMA>
    </OMA>
  </OMOBJ>
```

TABLE 5. A direct definition of softwarehouse's arctan

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
    <OMA>
      <OMS name="eq" cd="relation1"/>
      <OMA>
        <OMS name="arctan" cd="transc1" cdbase="http://www.softwarehouse.com/Derive"/>
        <OMV name="z"/>
      </OMA>
      <OMA>
        <OMS name="times" cd="arith1"/>
        <OMA>
          <OMS name="divide" cd="arith1"/>
          <OMS name="one" cd="alg1"/>
          <OMA>
            <OMS name="times" cd="arith1"/>
            <OMI> 2 </OMI>
            <OMS name="i" cd="nums1"/>
          </OMA>
        </OMA>
        <OMA>
          <OMS name="ln" cd="transc1"/>
          <OMA>
            <OMS name="divide" cd="arith1"/>
            <OMA>
              <OMS name="plus" cd="arith1"/>
              <OMS name="one" cd="alg1"/>
              <OMA>
                <OMS name="times" cd="arith1"/>
                <OMS name="i" cd="nums1"/>
                <OMV name="z"/>
              </OMA>
            </OMA>
            <OMA>
              <OMS name="minus" cd="arith1"/>
              <OMS name="one" cd="alg1"/>
              <OMA>
                <OMS name="times" cd="arith1"/>
                <OMS name="i" cd="nums1"/>
                <OMV name="z"/>
              </OMA>
            </OMA>
          </OMA>
        </OMA>
      </OMA>
    </OMA>
  </OMOBJ>
```

TABLE 6. Euclid's **P**

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
    <OMA>
      <OMS name="eq" cd="relation1"/>
      <OMS name="P" cd="CD" cdbase="http://www.euclid.gr/"/>
      <OMA>
        <OMS name="setdiff" cd="set1"/>
        <OMS name="N" cd="setname1"/>
        <OMA>
          <OMS name="set" cd="set1"/>
          <OMS name="zero" cd="alg1"/>
        </OMA>
      </OMA>
    </OMA>
  </OMOBJ>
```

TABLE 7. A definition of softwarehouse's arctan in `transc4`

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
    <OMA>
      <OMS name="eq" cd="relation1"/>
      <OMS name="arctan" cd="transc1" cdbase="http://www.softwarehouse.com/Derive"/>
      <OMS name="arctan" cd="transc4"/>
    </OMA>
  </OMOBJ>
```

TABLE 8. A definition of `Nstar`

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
<OMBIND>
    <OMS name="forall" cd="quant1"/>
    <OMBVAR>
       <OMV name="n"/>
    </OMBVAR>
    <OMA>
      <OMS name="implies" cd="logic1"/>
      <OMA>
        <OMS name="in" cd="set1"/>
        <OMV name="n"/>
        <OMS name="Nstar" cd="setname1"/>
      </OMA>
      <OMA>
        <OMS name="or" cd="logic1"/>
        <OMA>
          <OMS name="eq" cd="relation1"/>
          <OMV name="n"/>
          <OMS name="one" cd="alg1"/>
        </OMA>
        <OMA>
          <OMS name="in" cd="set1"/>
          <OMA>
            <OMS name="minus" cd="arith1"/>
            <OMV name="n"/>
            <OMS name="one" cd="alg1"/>
          </OMA>
          <OMS name="Nstar" cd="setname1"/>
        </OMA>
      </OMA>
    </OMA>
  </OMBIND>
</OMOBJ>
```

TABLE 9. A definition of `Landauin`

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0"
cdbase="http://www.openmath.org/cd">
  <OMA>
    <OMS cd = "logic1" name="implies"/>
    <OMA>
      <OMS cd = "asymp1" name="Landauin"/>
      <OMV name="A"/>
      <OMV name="B"/>
    </OMA>
    <OMA>
      <OMS cd = "set1" name="in"/>
      <OMV name="A"/>
      <OMV name="B"/>
    </OMA>
  </OMA>
</OMOBJ>
```