The Utility of OpenMath

James H. Davenport*

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

Abstract. OpenMath [5] is a standard for representing the *semantics* of mathematical objects. It differs from 'Presentation' MathML [7] in not being directly concerned with the presentation of the object, and from 'Content' MathML in being extensible. How should these extensions be performed so as to maximise the utility (which includes presentation) of OpenMath?

1 What is OpenMath?

"OpenMath is an emerging standard for representing mathematical objects with their semantics, allowing them to be exchanged between computer programs, stored in databases, or published on the worldwide web."[1]. In particular, OpenMath is extensible, unlike MathML 2.0[2] [7]. 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" [5, section 1.3].

Notation 1 *By an* OpenMath CD *we will mean any document conforming to the formal syntax of [5].*

The status of an OpenMath content dictionary is one of the following [5, Section 4.2.1]:

– `official`: approved by the *OpenMath* society according to the procedure defined in section 4.5 (of [5]);

* This paper owes much to some questions of Paul Libbrecht, when we were both 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 organising this workshop. Further comments, notably on section 6, are due to him [18] and Christian Gross [14]. Section 7 owes a lot to discussion with Prof. Vorobjov. Drs Naylor and Padegt also made useful suggestions.

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

[2] After this paper was submitted, a draft [8] of MathML 3.0 was produced, which bases content markup on OpenMath content dictionaries, and thus *is* extensible.

– `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. *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"/>
```

is a public symbol, whereas

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

is not.

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 [5, Section 4.4.2]. In the absence of such negotiation[5], 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). 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 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.

2 A Pragmatic Interpretation

Definition 2. *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.*

[3] This is the wording of [5]: the present author 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" [25]. For OpenMath, declaring a CD obsolete and writing a new one with the 'bug' fixed removes even this excuse: see section 6.

[5] Which may well be impossible in a "cut and paste" scenario.

Thus

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

appears to be a semi-public symbol, whereas

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

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 3. *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 ones that applications might reasonably encounter. This is what, effectively, is implied by the `cdbase` in the `OMOBJ` constructs quoted.

3 An example — arctan

One hypothetical example would be the following, for the system Derive[6], whose arctan function differs from the definition in [1]. As pointed out in [9], the two definitions could be related by the followingFMP.

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

[6] As already stated in [9], 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.

```
        <OMS name="arctan" cd="http://www.softwarehouse.com/Derive/transc1"/>
        <OMVAR name="z"/>
      </OMA>
      <OMA>
        <OMS name="conjugate" cd="complex1"/>
        <OMA>
          <OMS name="arctan" cd="transc1"/>
          <OMA>
            <OMS name="conjugate" cd="complex1"/>
            <OMVAR name="z"/>
          </OMA>
        </OMA>
      </OMA>
    </OMA>
  </OMOBJ>
</FMP>
```

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 as follows:

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

occurrences.

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 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 $\overset{\text{OpenMath}}{\longrightarrow}$ LaTeX would be distinctly surprised by the results, since the arctan emitted by 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 defined above. This has the disadvantage that the CD[7] 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 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.
   The interesting question is what an OpenMath⟶LaTeX phrasebook with no explicit Derive knowledge will do. It is unlikely to have the semantic processing capability to handle the FMP, though in this case it might. However,

[7] And the associated STS [11] file.

a plausible action by such a phrasebook would be to check the STS [11] file, observe that this function was unary from a set to itself (it might notice that the set was **C**, but this is irrelevant) and just print the name as a unary prefix function. Indeed, one could just observe that it was being used as a unary function, as is done in LeActiveMath [18, 24].

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 [5, 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

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

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) in terms of log, as OpenMath's arctan is in `transc1`.

Fig. 1. Definition of an alternative arctan

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

4 Another example

Let us imagine a theorem prover specialised in results 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 exports results such as "if the successor of a equals the successor of b, then $a = b$", i.e.

$$\forall a, b \in \mathbf{N}\, succ(a) = succ(b) \Rightarrow a = b? \tag{1}$$

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

1. Emit in terms of the OpenMath symbol, i.e. encode Euclid's $\mathbf{N}$ 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>
```

   This is certainly accurate, but would cause some grief on re-importing into Euclid, since:
   - $\mathbf{N}$ (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 $(\mathbf{N} \setminus \{0\}) \cup \{0\}$ is expecting rather more.
2. Emit in Euclid's own CD, e.g. with a definition as in figure 2. 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[8], normally rendered as $\mathcal{P}$. A configurable OpenMath→LaTeX converter[9] would be able to get this right, and print $\mathbf{P}$.
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 6.

[8] This is another example of the fact that an OpenMath symbol is the name *and* the CD.

[9] Such as the Notation Selection Tool [21, 22].

Fig. 2. Euclid's definition of **P** in terms of **N**

```
<FMP>
  <OMOBJ cdbase="http://www.openmath.org/cd">
    <OMA>
      <OMS name="eq" cd="relation1"/>
      <OMS name="P" cd="http://www.euclid.gr/CD"/>
      <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>
</FMP>
```

5 OpenMath and Notation

What use is OpenMath if one can't "see"[10] the results? Probably not much. How does one do it? One solution would be to make OpenMath do it.

> [...] was indicated as an expectation of Robert Miner at the W3C-Math f2f: if you find a CD, you should also have the notations with it ... so that you can present all the symbols in this CD. [18]

However, this begs the question: what is "the notation" [12]. 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[11] 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="arctan" cd="transc1"/>
```

the other is

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

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

[10] Used as shorthand for "convert into a presentation", which may be displayed in various means, e.g. audio [23].

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

To avoid the charge of antigallicanism being levied against the author, 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.

Hence it is impossible for an OpenMath object to know, in a context-free way, how it should be rendered[10]. 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 [6]. This would have the advantage of allowing technologies such as those described in [16, 23] to process it.

6 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[12], `transc1` is declared `obsolete`, `transc4` is promulgated with an FMP for arctan as in figure 1, and the authors of the softwarehouse CD change the FMP for arctan to be

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

and probably also mark that CD as `obsolete`. None of this would change the semantics of any OpenMath object.

However, the problem raised in section 4 is not so easily resolved: the question of whether $\mathbf{N}$ contains zero can, and indeed has [13], generate much debate. Many books, especially in German, suppose that $\mathbf{N}$ does *not* contain zero, e.g. the following.

[12] Caused by the requirement to move the branch cut in Network Time Protocol [20] 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".

> 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 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. [14]

It is clear that we have two different concepts, and several notations, as shown in Table 1.

Table 1. 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 figure 2, or we can define it in a free-standing way, by saying that it is 1 and its successors: formally

```
<OMOBJ 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>
```

(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.

7 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 author's). 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[13] textbooks then use '=', having apparently placated the god of Bourbaki[14]. However, actual uses of O as a set are rare: the author has never[15] seen "$O(f) \cap O(g)$", and, while a textbook might[16] 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 and transitive[17].

The set-theoretic view is the one taken by OpenMath CD[18] `asymp1`, except that only limiting behaviour at $+\infty$ is considered[19], 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: note 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 [17].
2. By attributing to each appropriate use of `<OMS name="in" cd="set1"/>` its print representation (at the moment there seems to be no standard way of doing this, though).
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 symbols such as `<OMS name="softO" cd="asymp2"/>`;
   (b) or for all usages of the (STS or other) type "function in set", thus printing $\sin = \mathbf{R}^{\mathbf{R}}$.

[13] [17] is an honourable exception.
[14] "the abuses of language without which any mathematical text threatens to become pedantic and even unreadable".
[15] Not even in the one context where it *would* be useful: $\Theta(f) = O(f) \cap \Omega(f)$, which is stated in words as [10, Theorem 3.1].
[16] [10, p. 41] write $\Theta(n) \subset O(n)$.
[17] Curiously enough, the FMPs currently only state transitivity: this probably ought to be fixed.
[18] Currently `experimental`.
[19] The CD author presumably considered that the level of abstraction needed for a more general definition was unwarranted. The current author would agree, especially as the context of O is generally only implicit in the wider context of the paper.

4. (the current author's favourite) By adding a symbol[20] `<OMS name="Landauin" cd="asymp1"/>`, which would, by default, print as '=', but have the semantics of '∈'.

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

```
<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 '∈' can become `Landauin`. A better way might be

```
<FMP>
<OMOBJ 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>
</FMP>
```

8 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

[20] It might be more appropriate to call it `Bachmannin`, since [4] is apparently the source of O. [15]

user, which means that OpenMath renderers *need* to be configurable, and at a variety of levels [19, section 4.2].

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.

8.1 Detailed suggestions

1. Add `<OMS cd = "asymp1" name="Landauin"/>`.
2. Add reflexive and symmetric properties to `<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 2 on page 12).

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. 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.
6. D.P. Carlisle. Openmath, MathML and XSLT. *ACM SIGSAM Bulletin 2*, 34:6–11, 2000.
7. 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/`.
8. World-Wide Web Consortium. Mathematical Markup Language (MathML) Version 3.0. *W3C Working Draft*, 2007. `http://www.w3.org/TR/2007/WD-MathML3-20070427`.
9. 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.
10. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to Algorithms, 2nd. ed.. *M.I.T. Press*, 2001.
11. J.H. Davenport. A Small OpenMath Type System. *ACM SIGSAM Bulletin 2*, 34:16–21, 2000.
12. J.H. Davenport. Nauseating Notation. `http://staff.bath.ac.uk/masjhd/Drafts/Notation.pdf`, 2007.
13. E.W. Dijkstra. Why numbering should start at zero. `http://www.cs.utexas.edu/users/EWD/transcriptions/EWD08xx/EWD831.html`, 1982.
14. C. Gross. Re: Utility of OpenMath. E-mail `64191.89.49.160.232.1172849601.squirrel@webmail.uni-augsburg.de`, 2007.

15. D.E. Knuth. Big Omicron and big Omega and big Theta. *ACM SIGACT News 2*, 8:18–24, 1974.
16. A. Lazrek. Multilingual Mathematical e-Document Processing. `http://www.ima.umn.edu/2006-2007/SW12.8-9.06/activities/Lazrek-Azzeddine/MathArabIMAe.pdf`, 2006.
17. A. Levitin. Introduction to the design and analysis of algorithms. *Pearson Addison-Wesley*, 2007.
18. P. Libbrecht. E-mails. `45B8875E.7000204@activemath.org`, 2007.
19. S. Manzoor, P. Libbrecht, C. Ullrich, and E. Melis. Authoring Presentation for OPENMATH. In M. Kohlhase, editor, *Proceedings MKM 2005*, pages 33–48, 2005.
20. D.L. Mills. Network Time Protocol, Version 3. `http://rfc.net/rfc1305.html`, 1992.
21. 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.
22. E. Smirnova and S.M. Watt. Notation Selection in Mathematical Computing Environments. In *Proceedings Transgressive Computing 2006*, pages 339–355, 2006.
23. N. Soiffer. Accessible Mathematics. `http://www.ima.umn.edu/2006-2007/SW12.8-9.06/activities/Soiffer-Neil/index.htm`, 2006.
24. Various. LeActiveMath. `http://www.activemath.org`, 2007.
25. D.J. Wheeler. Private Communication to J.H. Davenport. *June 1982*, 1982.