The Utility of OpenMath

James H. Davenport
Department of Computer Science
University of Bath
Bath BA2 7AY England
`J.H.Davenport@bath.ac.uk`

June 28, 2007

The status of an OpenMath content dictionary is one of the following:

- **`official`**: approved by the *OpenMath* society according to the procedure defined in section 4.5;
- **`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.

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.*

However, this is too restrictive

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.*

In practice, of course, people rely far more on experimental CDs than they "should", but that's practice for you.

Derive's arctan isn't OpenMath's

The designer of the Derive$\longrightarrow$OpenMath phrase-book is then faced with a set of alternatives.

1. Emit in terms of the public OpenMath symbol from `transc1`. If Derive can cancel double conjugation, it means that cut/paste from one Derive to another is not significantly more expensive. Some-one who is doing Derive$\xrightarrow{\text{OpenMath}}$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 a Derive symbol defined in a semi-public Derive CD. 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.

3. Ignore the problem, and emit `<OMS name="arctan" cd="transc1"/>`. Alas, this would be a very human reaction.

Either 1 or 2 is plausible.

Euclid thinks the natural numbers $\mathbf{N}$ are $1, \ldots$.

How should Euclid exports results such as

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

1. Emit in terms of the OpenMath symbol, i.e. encode Euclid's $\mathbf{N}$ as $\mathbf{N} \setminus \{0\}$. 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 of

   ```
   <OMS name="P" cd="http://www.euclid.gr/CD"/>
   ```

   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 $\mathcal{P}$, possibly confusing.

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.

OpenMath and Notation

- What use is OpenMath if one can't "see" the results?

- The "CD on the street" issue.

I like PL's idea of a `.ntn` file, which would presumably contain **a** conversion into MathML-P. A "search hierarchy' of `.ntn` files would allow for configuration by nationality, discipline etc.

OpenMath must (just) consider notation

- Everyone teaches that $T(n) = O(n^2)$ is an abuse of notation, and then (with one honourable exception), abuses it.

- This is certainly *not* `<OMS name="eq" cd="relation1"`

- **Semantically**, it certainly *is* `<OMS name="in" cd="set1"/>`.

So `<OMS name="Landauin" cd="asymp1"/>`, whose semantics are those of `<OMS name="in" cd="set1"/>`

Now, how is this rendered?

Presumably `asymp1.ntn` would have at least

```
<mo> = </mo>
```

producing $T(n) = O(n^2)$. This would unfortunately speak as “T of n equals big O of n squared”, which would make me do a rapid reality check on the student! At least in the U.K., I would expect “T of n is big O of n squared”.
Hence we would need *some* hint in the `.ntn` file.

AMC introduced a paper AMC/MK on "Defining Mathematical Properties". He said that CDs did not necessarily introduce the logical meaning of mathematical symbols. OpenMath should involve the logic community more. While OpenMath has Formal Mathematical Properties (FMPs), there is no differentiation between definitions and consequences. Also, some objects do not have FMPs, e.g. subset. He suggested a new tag, `DefMP`, which would be like FMPs, but the `DefMPs` would have to define the mathematical object uniquely. At least in theory, the FMPs would then be formally proved as consequences of the `DefMPs`.

In the Esprit group, there were two objections: one that they would scare many potential users, and the other was that peope might want different `DefMPs`. To the first, he answered that there were many features of OpenMath that not everyone used. For the second, he noted that signatures had been moved to separate files, and maybe this would be appropriate for `DefMPs`.

Types of OMS

1. Those that are fundamentally primitive, and not defined at all. They may still have FMPs, but these FMPs are merely about them, rather than defining the symbol. An example would be

   ```
   <OMS name="set" cd="set1"/>.
   ```

2. Those that OpenMath treats as primitive, and not defined at all in OpenMath. These might not be primitive in mathematics, but

OpenMath has decided not to define them. They may still have FMPs, but these FMPs are merely about them, rather than defining the symbol. An example would be

```
<OMS name="exp" cd="transc1"/>,
```

whose only FMP is a representation of $\forall k \in \mathbf{Z}\, \exp(z + 2k\pi i) = \exp(z)$ (which is equally true of $\exp(2z)$ for example).

3. At the other end of the spectrum, there are those objects that OpenMath defines (because mathematicians use them) but which

are logically redundant. An example of this is

```
<OMS name="sin" cd="transc1"/>,
```

whose FMP is a representation of $\sin(x) = \frac{\exp(ix)-\exp(-ix)}{2i}$, which means that all occurrences of `sin` can be removed from an OpenMath object without changing the semantics. *If the CD specified this*, a system which encountered a symbol like this could rewrite it knowing that there was no semantic loss.

If it felt that sin is still "important", and complex exponentials are not the right response to a real function, how about csc, which can be perfectly encapsulated via $\csc(x) = \frac{1}{\sin x}$?

4. It would be possible* (in fact the definition in `integer1` is not of this form, but rather in terms of products), to define

```
<OMS name="factorial" cd="integer1"/>
```

*If it is argued that this is artificial, since this is not in fact the FMP, consider the example of `Stirling1` in `combinat1`, whose FMP is the encoding of $Stirling1(n,m) = \sum_{k=0}^{n-m}(-1)^k * binomial(n-1+k, n-m+k) * binomial(2n-m, n-m-k) * Stirling2(n,m)$.

(whose STS states that it is a function $\mathbf{N} \to \mathbf{N}$) with an FMP encoding the recursive definition: In this case, it is possible to replace any particular numerical factorial by a computation, but it is impossible to replace, say $n!$ with a definition not involving factorials (unless one extracts some kind of Y-expression from that recursive definition, which is mere semantic trickery).

Therefore I now believe

That we should introduce a special kind of FMP, called `DefMP`, for *some* type 3,4 objects. This says "A is defined in terms of B" *and this is all there is to it*. In particular `LandauIn` would *not* have such a `DefMP`, since there *is* more, albeit only more notation, to it than `in`.