Defining Mathematical Properties

James Davenport [1]

7 July 2014

[1]Thanks to: Arjeh Cohen for first raising the question, many OpenMathers over the years, and Lars Hellström for "Varieties"

Background

OpenMath symbols `OMS` are defined in **Content Dictionaries** (CDs), user-specified and extensible
With symbols, they have CMPs (Commented Mathematical Properties), which are text, and FMPs (Formal Mathematical Properties), which are OpenMath
This paper is concerned purely with FMPs

A Little History

1999: AMC/MK "Defining Mathematical Properties" paper

- Various (contradictory) objections — see paper

2003: JHD "understands the issue better"

2007: MKM Further presentation

2008 Further inconclusive discussion

2014 This paper

(JHD's) four kinds 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. An example would be `<OMS name="exp" cd="transc1"/>`.
3. At the other end of the spectrum, there are those objects that OpenMath defines (because mathematicians use them) but which are logically redundant, such as sin or csc.
4. Symbols defined recursively in terms of themselves and more primitive objects, e.g. $n!$ as `n==0?1:n*(n-1)!`
   Hence 5! can be worked out, but not $n!$ (except in terms of fixed points/Y)

Type 2 Expanded

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 $2\exp(z)$ and $\exp(2z)$ for example).

Type3 Expanded

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 $\csc(x) = \frac{1}{\sin x}$?

Type 4 Expanded

It would be possible[2] (in fact the definition in `integer1` is not of this form, but rather in terms of products), to define

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

(whose STS states that it is a function $\mathbf{N} \rightarrow \mathbf{N}$) with an FMP encoding the recursive definition: `n==0?1:n*(n-1)!`
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).

[2] 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 $Stirling1(n, m) = \sum_{k=0}^{n-m}(-1)^k * binomial(n-1+k, n-m+k) * binomial(2n-m, n-m-k) * Stirling2(n, m)$.

The OpenMath dilemma

The notation of mathematics is incredibly varied, and new notations and concepts are permanently being introduced. This poses problems for OpenMath's goal of encouraging interoperability between tools, and future-proofing of data
Equally, people have different views of mathematics, e.g. Real Analysis/Complex Analysis, and this colours people's views of what is "fundamental"
In 2007, CSC distinguished three levels: notation (or presentation), content and logic. OpenMath, he thought, does well at distinguishing content from notation. He then asked whether `DefMP` wasn't mixing the last two — how can I interpret your `DefMP` if I don't know your logic? JHD admitted that this might be a problem for type 4 symbols. Type 3 symbols have a purely extensional definition, so the logic used should be irrelevant

Kinds of FMP: 2008 Proposal

At the moment, the distinction we have made above is purely informal, and there are no clues in the CD as to the meaning of any FMP. We now propose that some FMPs should be marked, and therefore *could* be treated specially.

More concretely, we propose two special marks: `type=`

defining A **defining** FMP is one that can always be used as a definition of a symbol. An example of this is the FMP for `sin` mentioned above. In all contexts, it is legitimate to replace an occurrence of `sin` by the corresponding right-hand side. Such FMPs will generally begin with an `eq` operator, though this is not necessarily required. The following guarantees must be met by such an FMP.

- A symbol can have at most one of them.
- The replacement value must not, either directly or indirectly by a chain of such FMPs, involve the symbol being defined.

Kinds of FMP: 2008 Proposal (continued)

evaluating An **evaluating** FMP is one that can be used as a definition of how to evaluate a symbol on a concrete instance of its input argument(s). The following guarantees must be met by such an FMP.

- A symbol can have at most one of them.
- The replacement value must, after a finite number of applications of this, and any other evaluating or defining FMPs, lead to an expression free of the symbol being defined, whenever the symbol is applied to concrete instances of the correct type(s).

The requirements for uniqueness: 2008

These requirements could be seen as posing the following questions.

1. Why restrict to one defining FMP?
2. Why restrict to one evaluating FMP?
3. Can one have one defining *and* one evaluating?

JHD: the first two avoid ambiguity — if I see two `defining`, which do I use?
JHD: the third is not so obvious: one would apply 'evaluating' where one could, else 'defining'

Varieties of Theory (Thanks, Lars)

There might be many competing view of what is "basic"
For functions we mght believe any of

C-Analysis sin in terns of exp

R-Analysis: 1 sin, cos primitive

R-Analysis: 2 $\sin(\theta)$ in terms of $\tan(\theta/2)$

A single `defining` cannot capture this variety

2014 proposal

As well as `type="defining"` or `type="evaluating"`, we allow `theories="theories"`, where theories is a list of theory identifiers. The uniqueness requirements are then interpreted *per theory*

Lars' further thoughts

"Also, if such names (I think the HTML class is of type `NMTOKENS`) can be namespaced, then that would probably be a nice way of associating the status of FMPs with theories. E.g. there could be

```
<FMP type="Euclid:definition Hilbert:axiom theorem">
```

to mean that "this FMP is known to be a definition in The Elements, an axiom according to Hilbert, and a theorem in some (unspecified) other theory". And the point about namespaces is that one would elsewhere have gone

```
<CD xmlns:Euclid="some-uri" xmlns:Hilbert="some-other-uri">
```

to provide a stable reference to "the theory", whatever that may be.I find the idea appealing that the URI for the Euclid theory should be an URL for The Elements, but that's probably a matter for DML/MKM to hash out. :-)"

Implications

Changes to

standard section 4.2, 4.3. Probably also add a clarification on the meaning of `type=` on the lines of this document

JHD believes that, since they are upwards-compatible, these could be in "OM2 second edition"

CDs To make use of this, we need to add some `type=` to existing CDs, and add more FMPs with this tag

version This would be a **minor** version number change

Compatibilty? In theory changing CD syntax is not upwards compatble, but JHD knows of no CD-reading tools except his students' units ones Any others?

Do we need a registry of theories? Lars' elsewhere

Or an index?