Binary Binding

An Extension for the OpenMath 2 Standard

MICHAEL KOHLHASE

School of Engineering & Sciences

International University Bremen, Germany

`http://www.cs.cmu.edu/~kohlhase`

Status Quo

- Currently: Unrestricted Binding

$\forall x.x=x$	$\lambda x,y.x(y)$

<pre><OMBIND>
  <OMS cd="quant1" name="forall"/>
  <OMBVAR><OMV name="x"/></OMBVAR>
  <OMA>
    <OMS cd="relation1" name="eq"/>
    <OMV name="x"/>
    <OMV name="x"/>
  </OMA>
</OMBIND></pre>
	<pre><OMBIND>
	  <OMS cd="fns1" name="lambda"/>
	  <OMBVAR>
	    <OMV name="x"/>
	    <OMV name="y"/>
	  </OMBVAR>
	  <OMA>
	    <OMV name="x"/>
	    <OMV name="y"/>
	  </OMA>
	</OMBIND></pre>

- But what about "$\forall n\in\mathbb{N}.n\geq 0$" (all natural numbers are non-negative)
- or "$\exists^1 n\in\mathbb{N}.prime(n)\wedge even(n)$" (there is exactly one even prime)
- or even "$\lambda x,y{:}\,x\neq y.\frac{1}{x-y}$" (the function where it is defined)
- or "almost all participants sleep"

But can't we do without?

- We have types in OpenMath isn't this enough?
  - Types are a decidable part of set theory built into the logic
  - a restriction mechanism can also be used for undecidable properties.
  - Example"$\lambda x{:}\,\zeta(x) \neq 0.x^2$ (we do not know where the zeros are)
- but surely we can relativize,
  - yes, sometimes, e.g. $\forall n \in \mathbb{N}.n \geq 0 \rightsquigarrow \forall n.n \in \mathbb{N} \Rightarrow n \geq 0$
  - but what about more complex quantifiers?

$$
\begin{aligned}
& \exists^1 n \in \mathbb{N}.prime(n) \wedge even(n) \\
= \quad & \exists n \in \mathbb{N}.prime(n) \wedge even(n) \wedge \forall m \in \mathbb{N} prime(n) \wedge even(n) \Rightarrow m = n \\
= \quad & \exists n.n \in \mathbb{N} \wedge prime(n) \wedge even(n) \wedge \forall m.(\mathtt{m} \in \mathbb{N} \wedge prime(n) \wedge even(n)) \Rightarrow m = n
\end{aligned}
$$

  - and what about functions? $\lambda x, y{:}\, x \neq y.\frac{1}{x-y} \rightsquigarrow \lambda x, y.\mathbf{if}\ x \neq y\ \mathbf{then}\ \frac{1}{x-y}$ leads to unwanted semantical constraints! (accepting axiom of choice)

IU$^{\mathbf{B}}$)

Concrete Proposals for Extension to OpenMath 2

- Idea: allow an optional “restriction element” as last child in <OMBVAR>

- Example: $\forall n \in \mathbb{N}.n \geq 0$

<table>
<tr><th>mark by element</th><th>mark by attribute</th></tr>
<tr><td>

```
<OMBIND>
 <OMS cd="quant1" name="forall"/>
 <OMBVAR>
  <OMV name="n"/>
  <OMRES>
   <OMA><OMS cd="set1" name="in"/>
    <OMV name="n"/>
    <OMS cd="setname1" name="N"/>
   </OMA>
  </OMRES>
 </OMBVAR>
 <OMA>
  <OMS cd="relation1" name="eq"/>
  <OMV name="x"/>
  <OMV name="x"/>
 </OMA>
</OMBIND>
```

</td><td>

```
<OMBIND>
 <OMS cd="quant1" name="forall"/>
 <OMBVAR restricted="yes">
  <OMV name="n"/>
  <OMA><OMS cd="set1" name="in"/>
   <OMV name="n"/>
   <OMS cd="setname1" name="N"/>
  </OMA>
 </OMBVAR>
 <OMA>
  <OMS cd="relation1" name="eq"/>
  <OMV name="x"/>
  <OMV name="x"/>
 </OMA>
</OMBIND>
```

</td></tr>
</table>

Conclusions

- need a restriction element for reasonable markup of complex binding structures (that is what you get, if you go away from CAS)
- will also help types (this is a welcome side effect)
- can be executed as a conservative extension
- ⚠ need to formulate the object model
- ⚠ need to adapt the binary encoding
- I really believe that we need this (not just causing trouble)

  IU^{B})