Structure Sharing in OpenMath

MICHAEL KOHLHASE

School of Computer Science
Carnegie Mellon University
Pittsburgh, USA

Fachbereich Informatik
Universität des Saarlandes
Saarbrücken, Germany

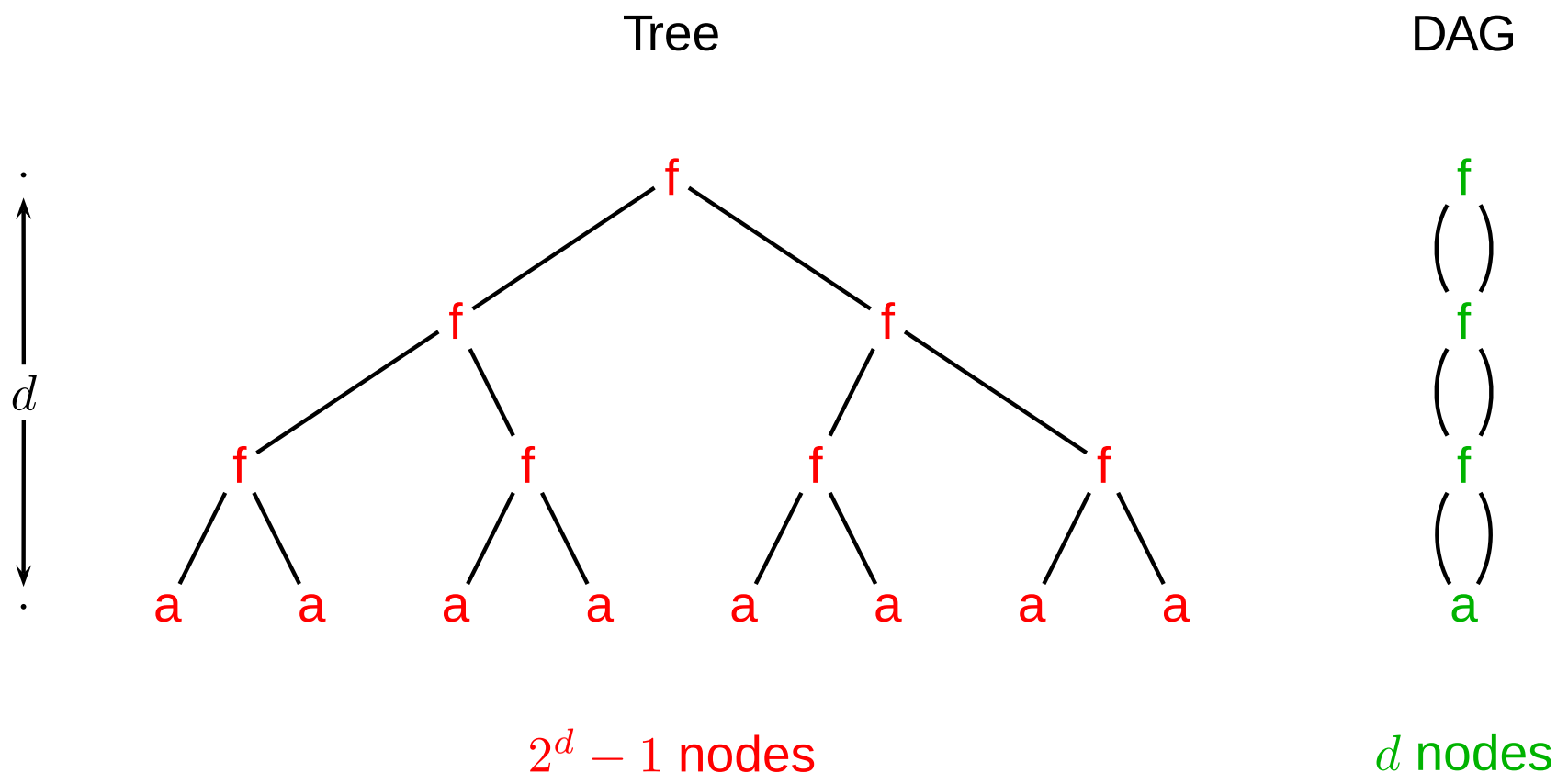
http://www.cs.cmu.edu/~kohlhase

(supported by the German Research Council under a Heisenberg grant)

Wish: Cross-refening for OpenMath objects

- Status: OMDoc just went ahead (licensed by the OpenMath standard)
  - new attributes `id` and `xref` for all OpenMath objects that carry content (not `OMS, OMV, OMATP`)
  - `OMel` with `xref` empty, `xref` points to element with same name
  - Semantics by copying. (simple transformation to standard OpenMath)
- Advantages: sharing of (sub)-formulae (+space, +maintainance)
- Problems: not based on XLink yet, semantics differs from MathML
- Proposal: Extend `OMel` with (cleaned up version for OpenMath)
  - `id` attribute (for OM `xref` (sharing), MathML `xref` (semantics))
  - `xref` attribute (for OM sharing)

Structure Sharing with Directed Acyclic Graphs



$2^d - 1$ nodes

d nodes

The same in the OpenMath XML encoding

```
<OMOBJ>
  <OMA>
    <OMV n="f"/>
    <OMA>
      <OMV n="f"/>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
    </OMA>
    <OMA>
      <OMV n="f"/>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
      <OMA>
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
    </OMA>
  </OMA>
</OMOBJ>
```

```
<OMOBJ>
  <OMA>
    <OMV n="f"/>
    <OMA id="t1">
      <OMV n="f"/>
      <OMA id="t11">
        <OMV n="f"/>
        <OMV n="a"/><OMV n="a"/>
      </OMA>
      <OMR xlink:href="t11"/>


    </OMA>
    <OMR xlink:href="t1"/>


  </OMA>
</OMA>
</OMOBJ>
```

Summary of the Proposal

- Idea: Allow structure sharing in the XML encodeing by
  - straw-man element `OMR` (represents target of `xlink:href` attribute)
  - by `id` attributes on "fat" OpenMath elements (possible targets)
- Pro: OpenMath data model does not change (stays finite trees.)

  Both encodings encode the OpenMath object

  application(f,**application**(f,**application**(f,a,a),
  application(f,a,a)),
  application(f,**application**(f,a,a),
  application(f,a,a)))
- Problem: Acyclicity Constraint (general DG represent infinite trees)

  non-local condition to be verified for validity

Acyclicity Condition

- Definition: We say that an
  - element dominates all its children and all elements they dominate.
  - An `OMR` element dominates its target, i.e. the element that carries the `id` attribute pointed to by the `xref` attribute.
- acyclicity constraint: An element may not dominate itself!
- Problem: Need to traverse the whole document tree to check.

Fun with Cyclic Graphs

- Cyclic data structures can be useful, e.g.

```
<OMOBJ>
  <OMA id="foo">
    <OMS cd="arith1" name="divide"/>
    <OMI>1</OMI>
    <OMA>
       <OMS cd="arith1" name="plus"/>
       <OMI>1</OMI>
       <OMR xref="foo"/>
    </OMA>
  </OMA>
</OMOBJ>
```

$$\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cdots}}}$$

- and difficult

```
<OMOBJ>
  <OMA id="bar">
    <OMS cd="arith1" name="plus"/>
    <OMI>1</OMI>
    <OMR xref="baz"/>
  </OMA>
</OMOBJ>
```

```
<OMOBJ>
  <OMA id="baz">
    <OMS cd="arith1" name="plus"/>
    <OMI>1</OMI>
    <OMR xref="bar"/>
  </OMA>
</OMOBJ>
```

Changes to the DTD

- add the declaration for the OMR element

```
<!ELEMENT OMR EMPTY>
<!ATTLIST OMR xlink:href CDATA #REQUIRED
              xlink:type CDATA #FIXED 'simple'
              xlink:show CDATA #FIXED 'embed'>
```

- add attribute list declarations `<!ATTLIST OMA id ID #IMPLIED>` for the elements `OMA`, `OMBIND`, `OMATTR`, `OMI`, `OMB`, `OMSTR`, `OMF`
  but not for elements
  - `OMS`, `OMV`, (to small, no need)
  - `OME`, `OMBVAR`, `OMATP`, (do not make sense on their own)
  - `OMOBJ` (`OMR` can only be used inside `OMOBJ`)
- extend the entity declaration for `%omel;`, so that it reads

```
<!ENTITY % omel "OMS | OMV | OMI | OMB | OMSTR| OMF |
                 OMA | OMBIND | OME | OMATTR | OMR">
```

A Synopsis of the Landscape of possible proposals

Proposal	DAGs in XML	DAG data struct.	CDGs
Acyclic?	yes/need check	yes/need check	no
Reader?	reader loops?	reader loops?	complex reader
Pros	same data model	model $\widehat{=}$ encoding	model $\widehat{=}$ encoding
motivation	save XML space	allow common DS	model complex DS
Legacy?	slightly extend reader	slightly extend DS	extend algorithms
character	conservative	innovative	radical