XSL Transformations

Simon Law

XSLT

- A W3C standard
- Used to transform XML
- Is designed to be used with XSL-FO
- Processor: xslproc and sablotron

XML to XML

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
                    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="text()|@*">
    <xsl:value-of select="normalize-space(.)"/>
  </xsl:template>
  <xsl:template match="text()|@*" mode="nonorm">
    <xsl:value-of select="."/>
  </xsl:template>
  <xsl:template match="/">
    <xsl:apply-templates/>
    <xsl:apply-templates mode="nonorm"/>
  </xsl:template>
</xsl:stylesheet>
```

XML to HTML

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
                      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" indent="yes"/>
  <xsl:template match="text()|@*">
    <xsl:value-of select="normalize-space(.)"/>
  </xsl:template>
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml"
             xml:lang="en" lang="en">
      <head><title>Web</title></head>
      <body><xsl:apply-templates/></body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

XML to LaTeX

```
<xsl:template name="substitute">
  <xsl:param name="input"/>
  <xsl:param name="old"/>
  <xsl:param name="new"/>
  <xsl:choose>
    <xsl:when test="contains($input, $old)">
      <xsl:value-of select="substring-before($input, $old)"/>
      <xsl:value-of select="$new"/>
      <xsl:call-template name="substitute">
        <xsl:with-param name="input">
          <xsl:value-of select="substring-after($input, $old)"/>
        </xsl:with-param>
        <xsl:with-param name="old">
          <xsl:value-of select="$old">
        </xsl:with-param>
        <xsl:with-param name="name">
          <xsl:value-of select="$new"/>
        </xsl:with-param>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="$input"/>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

XML to LaTeX

```
<xsl:template name="escape-specials">
  <xsl:param name="input"/>
  <xsl:call-template name="substitute">
    <xsl:with-param name="input">
      <xsl:call-template name="substitute">
        <xsl:with-param name="input">
          <xsl:value-of select="$input"/>
        </xsl:with-param>
        <xsl:with-param name="old">\</xsl:with-param>
        <xsl:with-param name="new">\textbackslash </xsl:with-param>
      </xsl:call-template>
    </xsl:with-param>
    <xsl:with-param name="old">$</xsl:with-param>
    <xsl:with-param name="new">\$</xsl:with-param>
  </xsl:call-template>
</xsl:template>

<xsl:template match="text()|@*">
  <xsl:call-template name="escape-specials">
    <xsl:with-param name="input">
      <xsl:value-of select="normalize-space(.)"/>
    </xsl:with-param>
  <xsl:call-template>
</xsl:template>
```

Scheme

- s-expressions
- Scheme can be used to match XML
- SXML