Create SVG in XSL FO programmatically. Usecase #1: tree-like structures

XSL FO allows you to embed different types of graphic in your documents. One of such types is SVG:

Scalabe Vector Graphics (SVG)

Apart from common vector graphics advantages (such as no-loss quality and relatively small size for certain drawing types) SVG is an XML itself. Thus you can not only embed images produced by external applications, but create SVG programmatically from arbitrary XML data by means of XSLT. If you have some private data in XML you can present them both in textual and graphical forms using single stylesheet that produce XSL FO and SVG.

Below we will demonstrate one of possible usecases – creation of SVG representation of tree-like structure from XML data. This document contains sentence analysis in the form of the following XML:

```
<S>
  <NP>
    <Det>The</Det>
    <NP>
      <Adj>quick</Adj>
      <NP>
        <Adj>brown</Adj>
        <N>fox</N>
      </NP>
    </NP>
  </NP>
  <VP>
    <V>jumps</V>
    <PP>
      <Prep>over</Prep>
      <NP>
        <Det>a</Det>
        <NP>
          <Adj>lazy</Adj>
          <N>dog</N>
        </NP>
      </NP>
    </PP>
  </VP>
</S>
```

Dedicated XSLT stylesheet produces both textual (bracket structure) and graphical (tree-like) representation of this data as shown below. XSLT stylesheet contains a number of templates, which matches elements mentioned above and produce SVG elements (placed in appropriate SVG namespace `xmlns:svg="http://www.w3.org/2000/svg"`) such as lines and text labels. As example, SVG code fragment produced by stylesheet cited below:

```
    <fo:instream-foreign-object width="100%"
                                content-width="scale-to-fit"
                                content-height="100%">
      <svg:svg width="270pt"
               height="170pt"
               viewBox="0 0 270 160">
          ...
        <svg:text style="font-family:Helvetica; font-size:10"
                  text-anchor="middle"
                  x="135" y="20">S</svg:text>
        <svg:line style="stroke-width:2; stroke: darkred; stroke-linecap:
                  x1="135" y1="23" x2="60" y2="30"/>
          ...
      </svg:svg>
    </fo:instream-foreign-object>
```

Sentence analysis

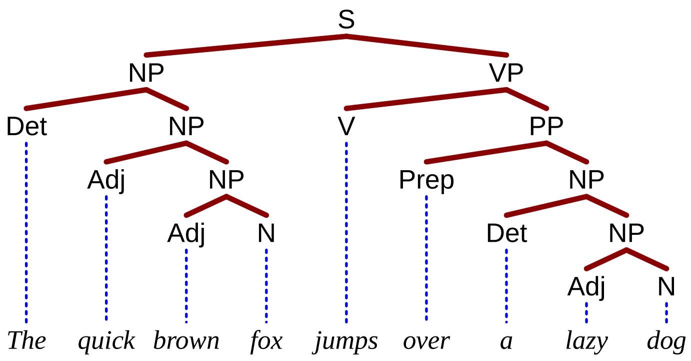
Sentence1:

The quick brown fox jumps over a lazy dog

Bracket structure:

[[*The* $^{\text{Det}}$[*quick* $^{\text{Adj}}$[*brown* $^{\text{Adj}}$ *fox* $^{\text{N}}$]$^{\text{NP}}$]$^{\text{NP}}$]$^{\text{NP}}$[*jumps* $^{\text{V}}$[*over* $^{\text{Prep}}$[*a* $^{\text{Det}}$[*lazy* $^{\text{Adj}}$ *dog* $^{\text{N}}$]$^{\text{NP}}$]$^{\text{NP}}$]$^{\text{PP}}$]$^{\text{VP}}$]$^{\text{S}}$

Tree structure:


S
NP
VP
Det
NP
V
PP
Adj
NP
Prep
NP
Adj
N
Det
NP
Adj
N
The
quick
brown
fox
jumps
over
a
lazy
dog