Data Deduplication

Amanda Buckley & Jayce Gaines

Introduction (1)

What is Data De-Duplication?

– One file with many copies on the disk.

– Redundant information within files which are not necessarily the same

Why is it important?

– Disk space is expensive.

– Multiple copies can diverge over time, creating inconsistencies in the data.

Introduction (2)

What is structured text?

– Stores data items and relationships between them
– Data is stored in plain text, marked up with tags
– Forms a tree structure
– Difficult to de-duplicate

What is XML?

- XML - eXtensible Markup Language.
- A language to structure, store and transport data.
- Human readable.
- Schema extendable with namespaces
- Used for graphics, news feeds, word documents

Applications of XML

- Web pages
- Really Simple Syndication (RSS) and Atom Feeds
  - Specified Formats
- Office Documents
  - OpenOffice.org XML
  - Microsoft's .docx format
- Scalable Vector Graphics (SVG) Files
  - Language for 2D drawings in XML

XML Nodes

- Elements
- Attributes
- Entities
- Processing Instructions
- Comments
- CDATA Sections

XML Data De-Duplication

- Goal: create a software library and accompanying application for finding the difference between two XML input files
- Output the result as a parseable XML file
- Structured text is hard to de-duplicate
- Tree structure does not depend on line order
- Files contain meta-information as well as data
- <hr/> is the same as <hr></hr>

Background (1)

File System De-Duplication

– De-duplicates data on the fly over the network

– Uses hashing or other sophisticated data structure techniques to find duplicate blocks within files

– Does not solve data inconsistency problem

Background (2)

De-duplication Utilities

- Unix “diff” command
  - Outputs differences between two files
  - Operates line by line
  - Not suitable for tree structures
  - Not suitable for binary data

- OpenXMLDiff
  - Command-line program, pipes output to text file

Background (3)

Diffxml utility

– Doctoral Dissertation by Adrian Mouat

– Outputs diff in “DUL” (Delta Update Language)

– Written in Java

– Open Source, but not available as a library

– Limited to small files

Xmldiff

– Open Source Python script

– Can be used as a library

– Only documentation is a French blog post???

Background (4)

Proprietary Tools

- DiffDog
  - Provides diff/merge for text files and ODTs
  - “XML-aware” approach to visualization
  - Provides different options for customization
- DeltaXML Ltd's “Delta XML”
- “XML Diff and Merge” and “XML Tree Diff”

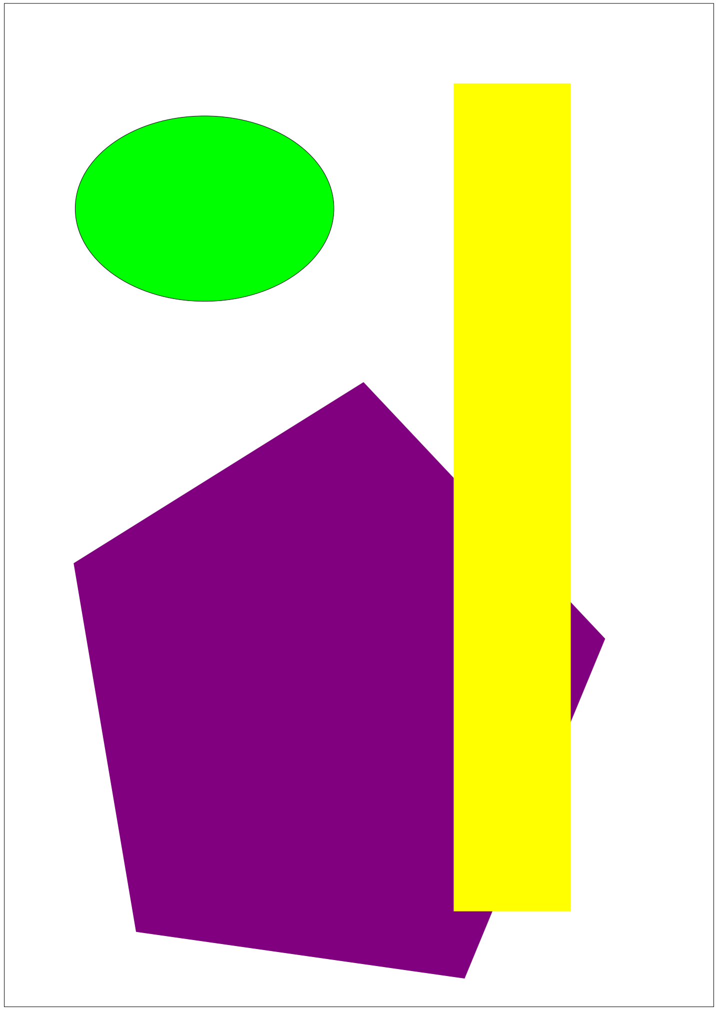
Background (5)

- DiffDog
  - Provides diff/merge for text files and ODTs
  - “XML-aware” approach to visualization
  - Provides different options for customization
  - Proprietary
- OpenXMLDiff
  - Free
  - Command-line program, pipe output to text file

FastXMLDiff

- Open Source program for finding differences between two XML trees
- Outputs the UNION of the two trees with appropriate annotations
- This allows it to produce files that can be opened by the application that created the files
- Cross-platform GUI app based on Qt
- Could easily be made into a library for use in other applications

Example Image

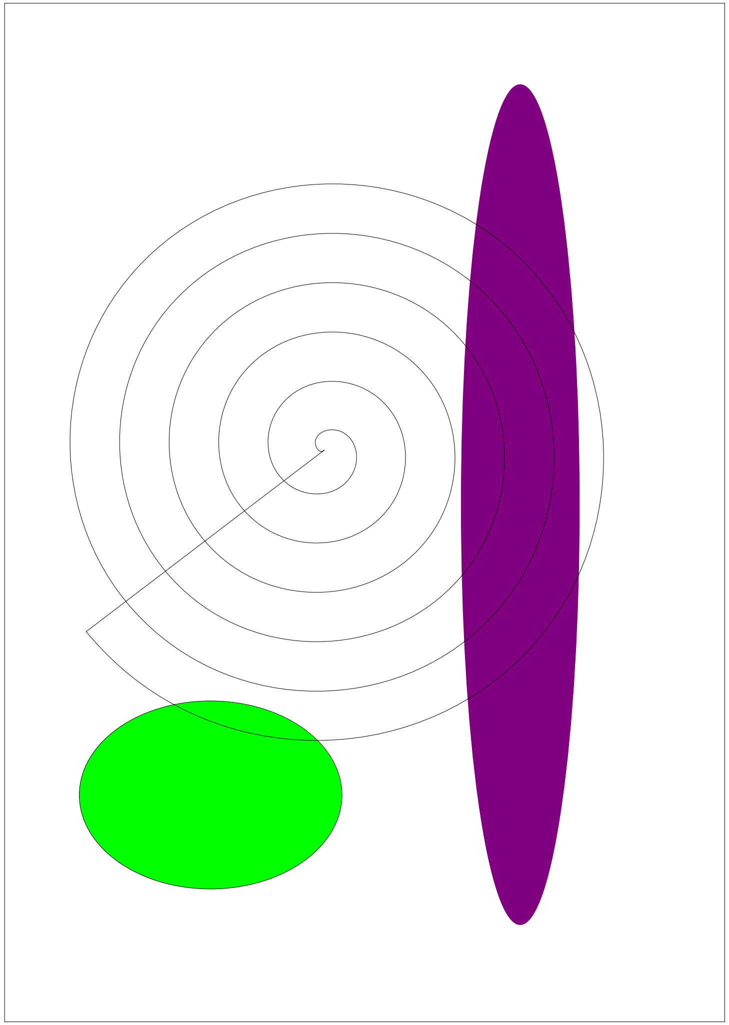


Example Document (XML)

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>

<svg width="744.09448819"
   height="1052.3622047"
   id="svg2"
   version="1.1"
   inkscape:version="0.48.0 r9654"
   sodipodi:docname="testimage1.svg">

  <sodipodi:namedview id="base"
     pagecolor="#ffffff"
     bordercolor="#666666"
     borderopacity="1.0"
     inkscape:pageopacity="0.0"
     inkscape:pageshadow="2"
     inkscape:zoom="0.35"
     inkscape:cx="375"
     inkscape:cy="520"
     inkscape:document-units="px"
     inkscape:current-layer="layer1"
     showgrid="false"
     inkscape:window-width="1280"
     inkscape:window-height="947"
     inkscape:window-x="0"
     inkscape:window-y="24"
     inkscape:window-maximized="1" />

  <g inkscape:label="Layer 1"
     inkscape:groupmode="layer"
     id="layer1">
```

```
<path id="path2985"
    style="fill:#00ff00;fill-
rule:evenodd;stroke:#000000;stroke-
width:1px;stroke-linecap:butt;stroke-
linejoin:miter;stroke-opacity:1"
d="m 345.71428,215.21933 a
135.71428,97.14286 0 1 1 -271.428559,0
135.71428,97.14286 0 1 1 271.428559,0 z" />

<path style="fill:#800080”
 id="path3011"
 inkscape:flatsided="true"
 inkscape:rounded="0"
 inkscape:randomized="0"
 d="M 114.28572,720.93363
72.695787,503.89155 266.26307,397.26748
427.48416,548.41226 333.557,748.44893 z"
inkscape:transform-center-x="-11.368332"
inkscape:transform-center-y="-19.471307"
transform="matrix(1.5717694,0,0,1.781037,-
41.565228,-310.28061)" />
<rect
       style="fill:#ffff00"
       id="rect3140"
       width="122.85714"
       height="868.57141"
       x="471.42856"
       y="83.790756" />
</g>

</svg>
```

Example Image (Modified)



Modified Example Document:

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<svg width="744.09448819"
   height="1052.3622047"
   id="svg2" version="1.1"
   inkscape:version="0.47 r22583"
sodipodi:docname="testimage2.svg">
  <sodipodi:namedview id="base"
     pagecolor="#ffffff"
     bordercolor="#666666"
     borderopacity="1.0"
     inkscape:pageopacity="0.0"
     inkscape:pageshadow="2"
     inkscape:zoom="0.35"
     inkscape:cx="375"
     inkscape:cy="514.28571"
     inkscape:document-units="px"
     inkscape:currentlayer="layer1"
     showgrid="false"
     inkscape:window-width="1280"
     inkscape:window-height="949"
     inkscape:window-x="0"
     inkscape:window-y="25"
     inkscape:window-maximized="1" />
<g inkscape:label="Layer 1"
 inkscape:groupmode="layer"id="layer1">
<path style="fill:#00ff00;fill-
rule:evenodd;stroke:#000000;stroke-
width:1px;stroke-linecap:butt;stroke-
linejoin:miter;stroke-opacity:1"
id="path2985" d="m 345.71428,215.21933 a
135.71428,97.14286 0 1 1 -271.428559,0
135.71428,97.14286 0 1 1 271.428559,0 z"
transform="translate(2.8571433,602.85714)" />
```

```
<rect style="fill:#800080" id="rect3140"
     Width="122.85714" Height="868.57141"
     X="471.42856" y="83.790756" ry="61.42857" />
<path sodipodi:type="spiral"
style="fill:none;stroke:#000000;stroke-
width:1px;stroke-linecap:butt;stroke-
linejoin:miter;stroke-opacity:1"
id="path3142" d="m 340,166.6479 c -3.16672,3.30861
-6.02228,-2.63632 -5.49912,-5.26329 1.41773,-7.11893
10.46709,-8.45664 16.02571,-5.73495 9.94306,4.86846
11.38412,18.045 5.97078,26.78812 -7.9443,12.83089
-25.72474,14.39569 -37.55054,6.20661 -15.76193,-
10.91475 -17.44292,-33.44049 -6.44244,-48.31295
13.84297,-18.71547 41.17339,-20.50856 59.07537,-
6.67827 21.68169,16.75032 23.58485,48.91593
6.9141,69.83778 -19.64591,24.65568
-56.66449,26.66784 -80.6002,7.14993 -27.63473,-
22.53419 -29.75529,-64.41705 -7.38576,-91.36261
25.41768,-30.617257 72.17244,-32.845862 102.12503,-
7.62159 33.60227,28.29782 35.9387,79.92988
7.85742,112.88744 -31.17554,36.58914
-87.68888,39.03324 -123.64986,8.09325 -39.5774,-
34.05146 -42.12907,-95.44906 -8.32908,-134.412271
36.926,-42.566747 103.21018,-45.225933 145.17469,-
8.564911 45.55695,39.799462 48.32361,110.972072
8.80074,155.937102 -42.67206,48.54786
-118.73457,51.42195 -166.69952,9.03657 -51.53933,-
45.54395 -54.52083,-126.49757 -9.2724,-177.461932
48.41528,-54.531269 134.261,-57.620159 188.22435,-
9.50823 57.5236,51.286132 60.71987,142.024792
9.74406,198.986762 -54.15659,60.51627
-149.78888,63.81992 -209.74918,9.97989 -63.50922,-
57.02669 -66.92023,-157.55323 -10.21572,-220.511592
59.89651,-66.502409 165.31781,-70.020778 231.27401,-
10.45155 69.49581,62.766082 73.12154,173.082592
10.68738,242.036422 -65.63543,72.4894
-180.84754,76.22248 -252.79884,10.92321 -6.98202,-
6.33651 -13.47185,-13.21429 -19.39508,-20.55012"
  transform="matrix(1.6895195,0,0,1.6840156,-
243.94779,180.82631)" />
  </g>
</svg>
```

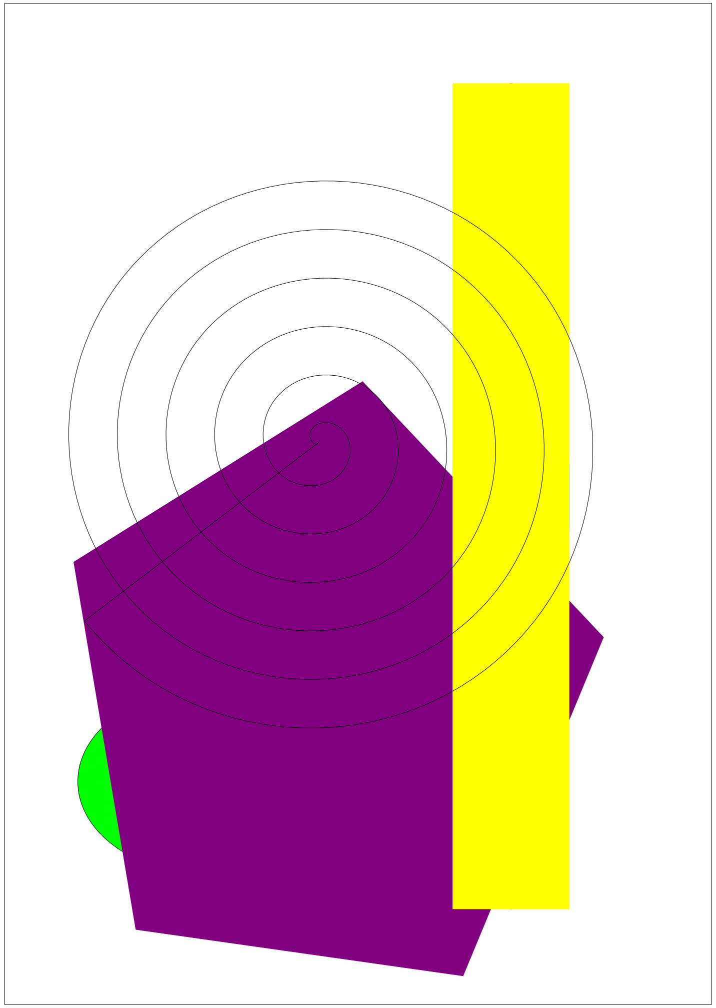
Example Output

```
<root>
<?xml version='1.0' encoding='UTF-8' standalone='no'?>
 <svg
   width="744.09448819"
   version="1.1"
   height="1052.3622047"
   xmlns:sodipodi="http://sodipodi.sourceforge.net/DTD/sodipodi-0.dtd"
   sodipodi:docname="testimage1.svg"
   id="svg2"
   xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape">
  <sodipodi:namedview
   inkscape:window-y="25"
   inkscape:window-maximized="1"
   inkscape:zoom="0.35"
   inkscape:document-units="px"
   showgrid="false"
   pagecolor="#ffffff"
   bordercolor="#666666"
   inkscape:cx="375"
   id="base"
   inkscape:cy="520"
   inkscape:window-height="949"
   inkscape:pageopacity="0.0"
   borderopacity="1.0"
   inkscape:pageshadow="2"
   inkscape:current-layer="layer1"
   inkscape:window-width="1280"
   inkscape:window-x="0"/>
  <g
   inkscape:label="Layer 1"
   inkscape:groupmode="layer"
   id="layer1">
   <path
   modified="true"
   style="fill:#00ff00;fill-rule:evenodd;stroke:#000000;stroke-
width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1"
   id="path2985"
   d="m 345.71428,215.21933 a 135.71428,97.14286 0 1 1
-271.428559,0 135.71428,97.14286 0 1 1 271.428559,0 z"
   transform="translate(2.8571433,602.85714)"/>
```

```
 <container-node
    modified="true">
    <path
     inkscape:transform-center-x="-11.368332"
     inkscape:rounded="0"
     inkscape:transform-center-y="-19.471307"
     inkscape:flatsided="true"
     style="fill:#800080"
     id="path3011"
     inkscape:randomized="0"
     d="M 114.28572,720.93363 72.695787,503.89155
266.26307,397.26748 427.48416,548.41226 333.557,748.44893 z"
     transform="matrix(1.5717694,0,0,1.781037,-41.565228,-
310.28061)"/>
    <rect
     width="122.85714"
     x="471.42856"
     y="83.790756"
     height="868.57141"
     ry="61.42857"
     style="fill:#800080"
     added="true"
     id="rect3140"/>
   </container-node>
   <container-node
    modified="true">
    <rect
     width="122.85714"
     x="471.42856"
     y="83.790756"
     height="868.57141"
     style="fill:#ffff00"
     id="rect3140"/>
```

Example output (cont.)

```
<path
     sodipodi:type="spiral"
     style="fill:none;stroke:#000000;stroke-width:1px;stroke-
linecap:butt;stroke-linejoin:miter;stroke-opacity:1"
     added="true"
     id="path3142"
     d="m 340,166.6479 c -3.16672,3.30861 -6.02228,-2.63632 -5.49912,-
5.26329 1.41773,-7.11893 10.46709,-8.45664 16.02571,-5.73495
9.94306,4.86846 11.38412,18.045 5.97078,26.78812 -7.9443,12.83089
-25.72474,14.39569 -37.55054,6.20661 -15.76193,-10.91475
-17.44292,-33.44049 -6.44244,-48.31295 13.84297,-18.71547
41.17339,-20.50856 59.07537,-6.67827 21.68169,16.75032
23.58485,48.91593 6.9141,69.83778 -19.64591,24.65568
-56.66449,26.66784 -80.6002,7.14993 -27.63473,-22.53419 -29.75529,-
64.41705 -7.38576,-91.36261 25.41768,-30.617257 72.17244,-
32.845862 102.12503,-7.62159 33.60227,28.29782 35.9387,79.92988
7.85742,112.88744 -31.17554,36.58914 -87.68888,39.03324
-123.64986,8.09325 -39.5774,-34.05146 -42.12907,-95.44906 -8.32908,-
134.412271 36.926,-42.566747 103.21018,-45.225933 145.17469,-
8.564911 45.55695,39.799462 48.32361,110.972072
8.80074,155.937102 -42.67206,48.54786 -118.73457,51.42195
-166.69952,9.03657 -51.53933,-45.54395 -54.52083,-126.49757
-9.2724,-177.461932 48.41528,-54.531269 134.261,-57.620159
188.22435,-9.50823 57.5236,51.286132 60.71987,142.024792
9.74406,198.986762 -54.15659,60.51627 -149.78888,63.81992
-209.74918,9.97989 -63.50922,-57.02669 -66.92023,-157.55323
-10.21572,-220.511592 59.89651,-66.502409 165.31781,-70.020778
231.27401,-10.45155 69.49581,62.766082 73.12154,173.082592
10.68738,242.036422 -65.63543,72.4894 -180.84754,76.22248
-252.79884,10.92321 -6.98202,-6.33651 -13.47185,-13.21429
-19.39508,-20.55012"
     transform="matrix(1.6895195,0,0,1.6840156,-
243.94779,180.82631)"/>
   </container-node>
  </g>
 </svg>
</root>
```



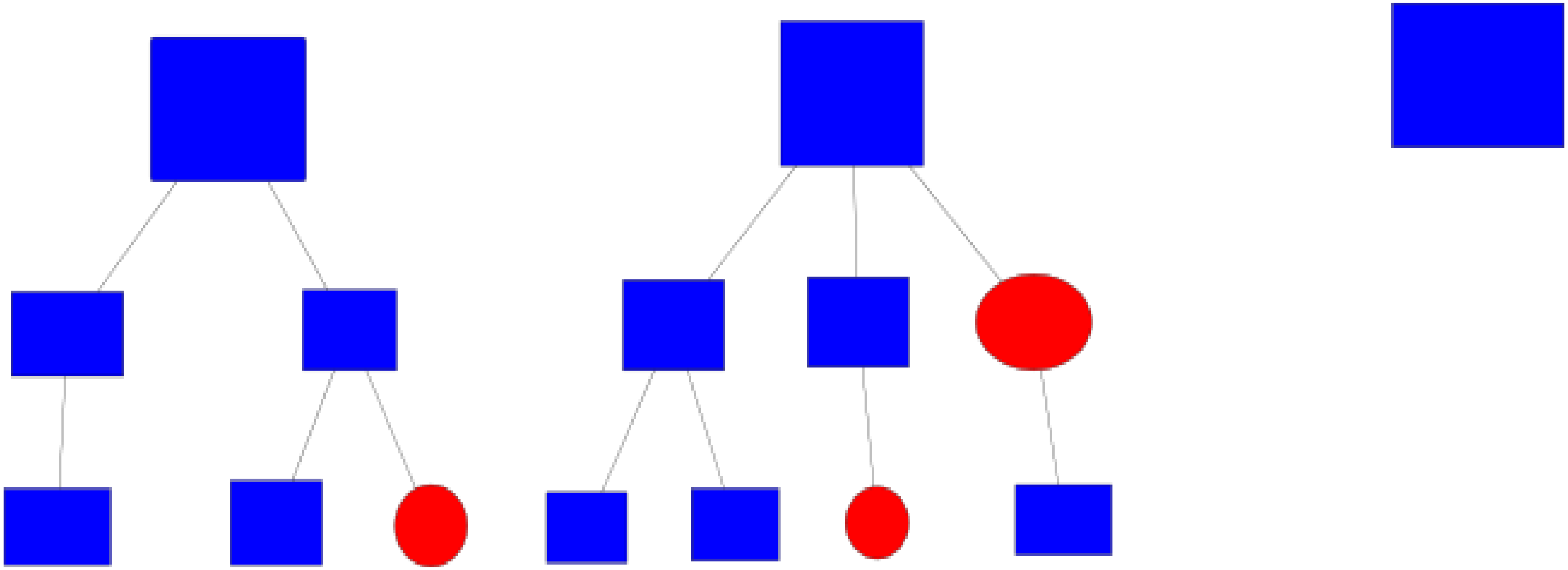
Algorithm (1)

- Creates a new tree representing the union of the two files.
- Annotates the new tree by marking nodes as:
  - Modified
  - Added
  - Deleted
- Uses a recursive tree-union algorithm.
- Comparing two nodes
  - Comparing Elements
  - Comparing Entities and other non-elements
  - Handling attributes

Algorithm (2)

```
XmlTree CompareXML(XmlTree A, XmlTree B):
   root = new_node()
   for each child a[i] in A:
      if ∃ b[i] in B:
      if a[i] ≠ b[i]:
         create a container node “c”.
         append a[i] and b[i] to c.
         mark “c” as modified.
         append “c” to root.
      else
         make a copy “c” of a[i].
         mark “c” as deleted.
         append “c” to root.
   for each child b[i] in B\A:
      make a copy “c” of b[i].
      mark “c” as added.
      append “c” to root.
return root
```

Algorithm (3)

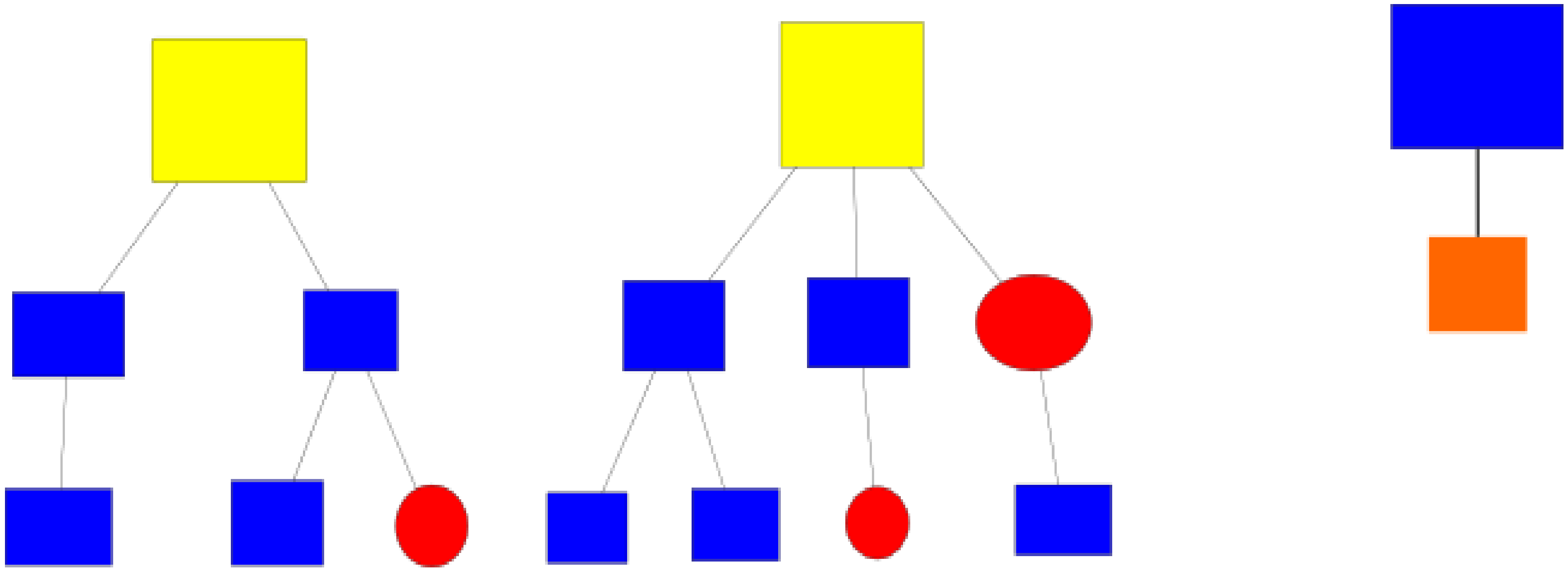


Old

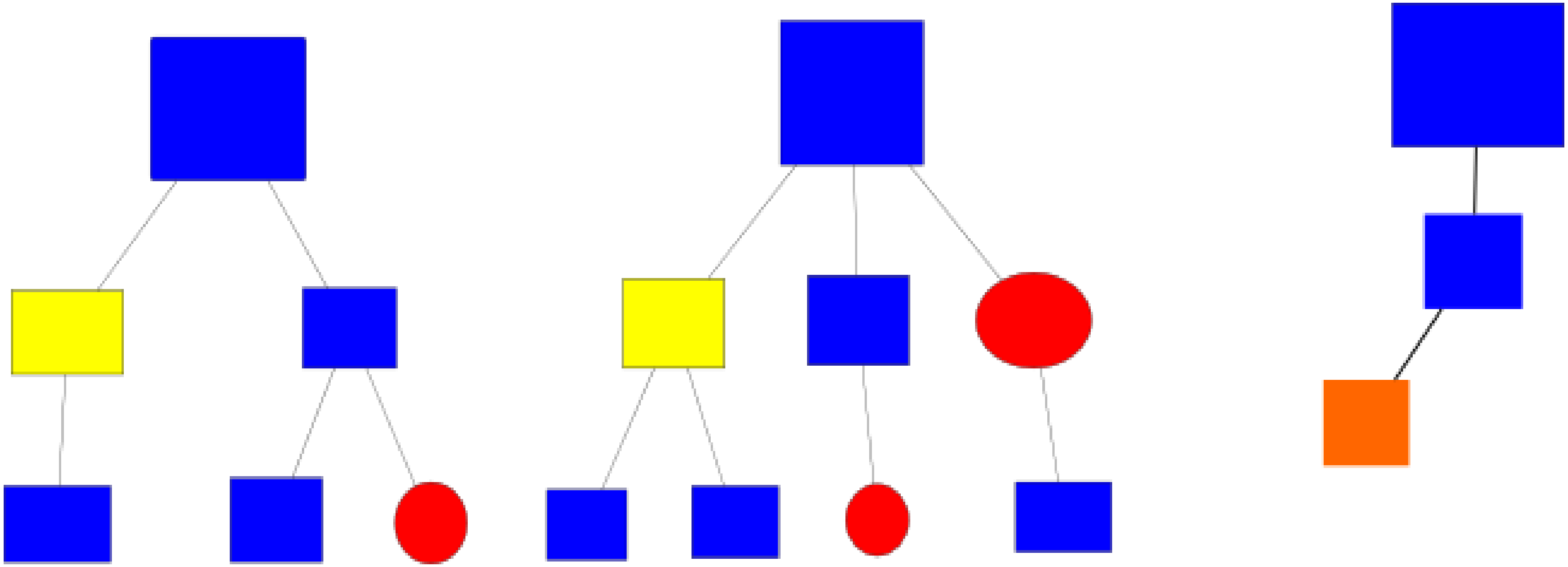
New

Result

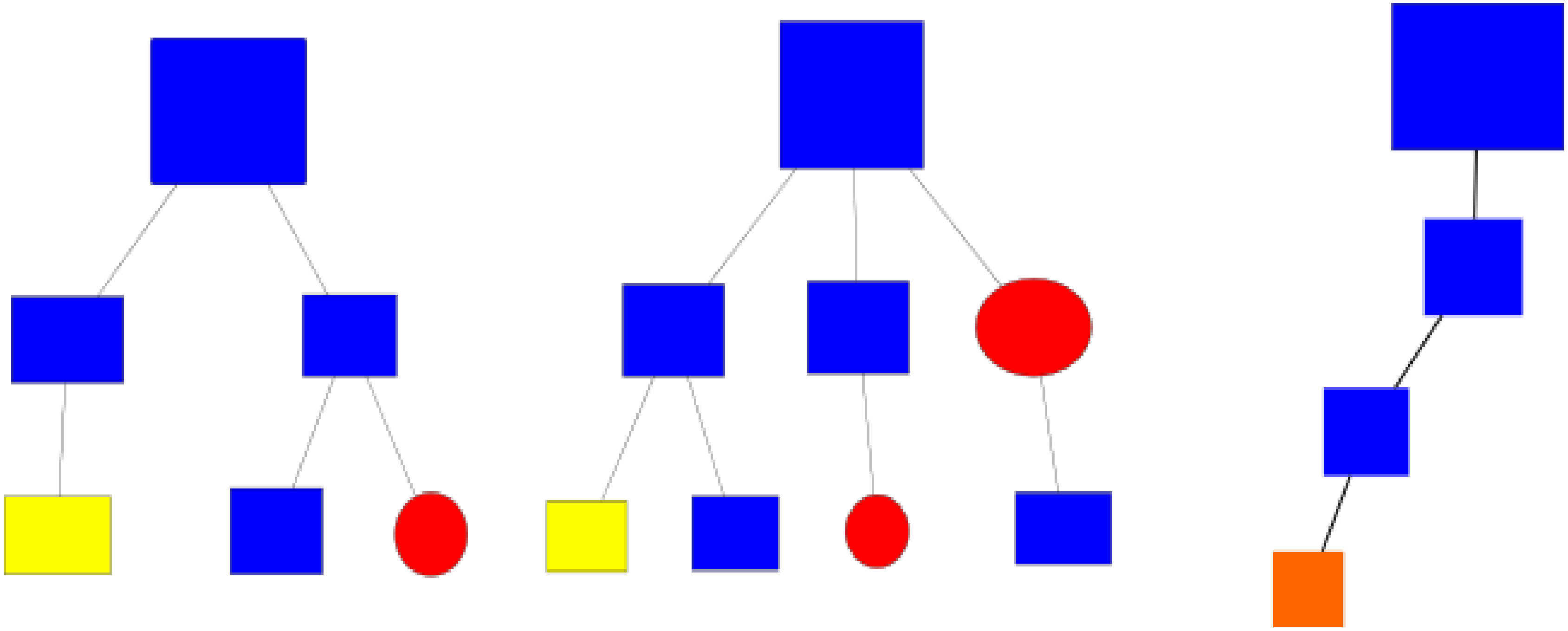
Algorithm (4)



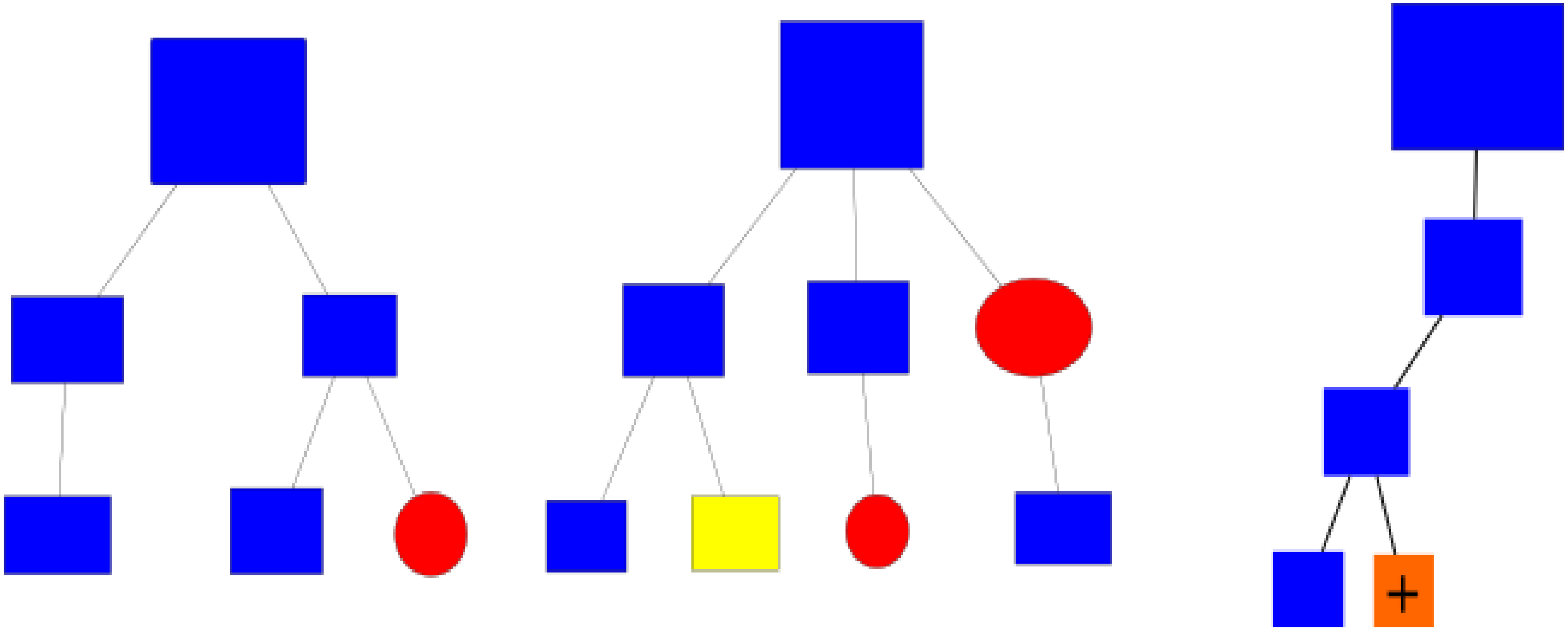
Algorithm (5)



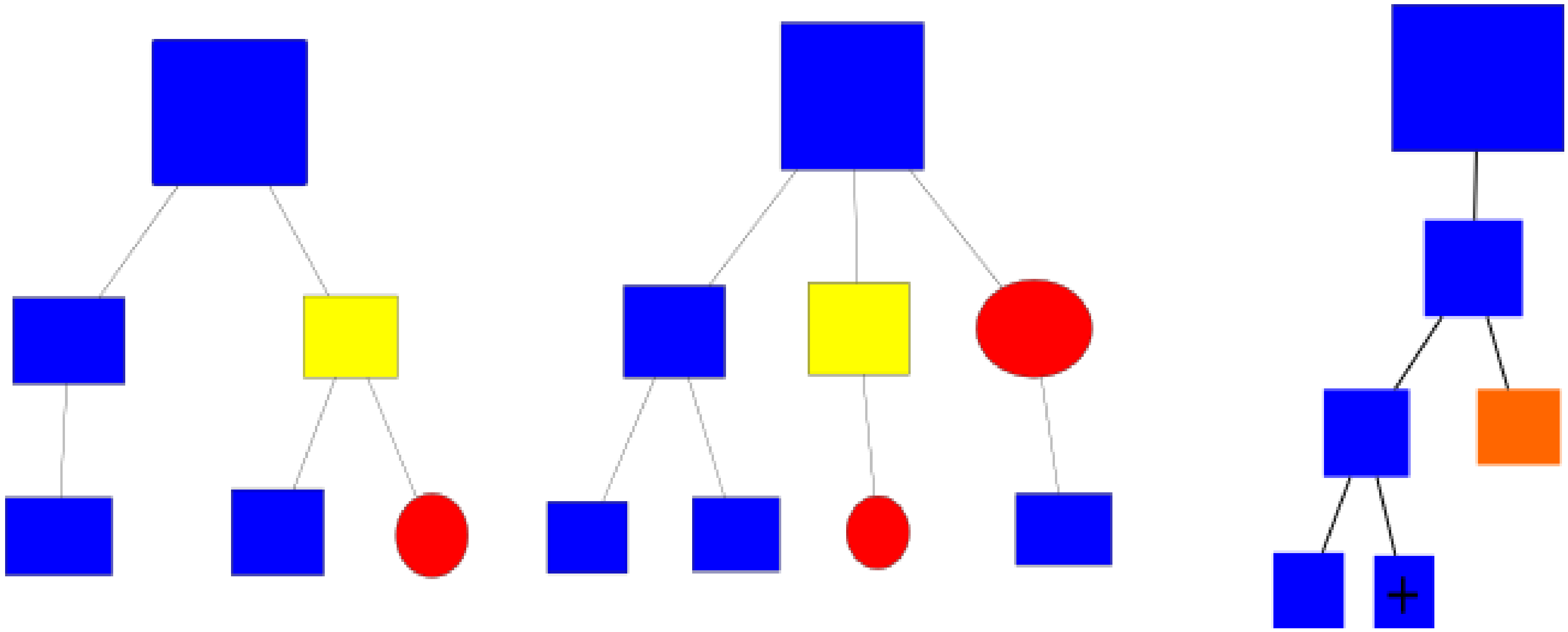
Algorithm (6)



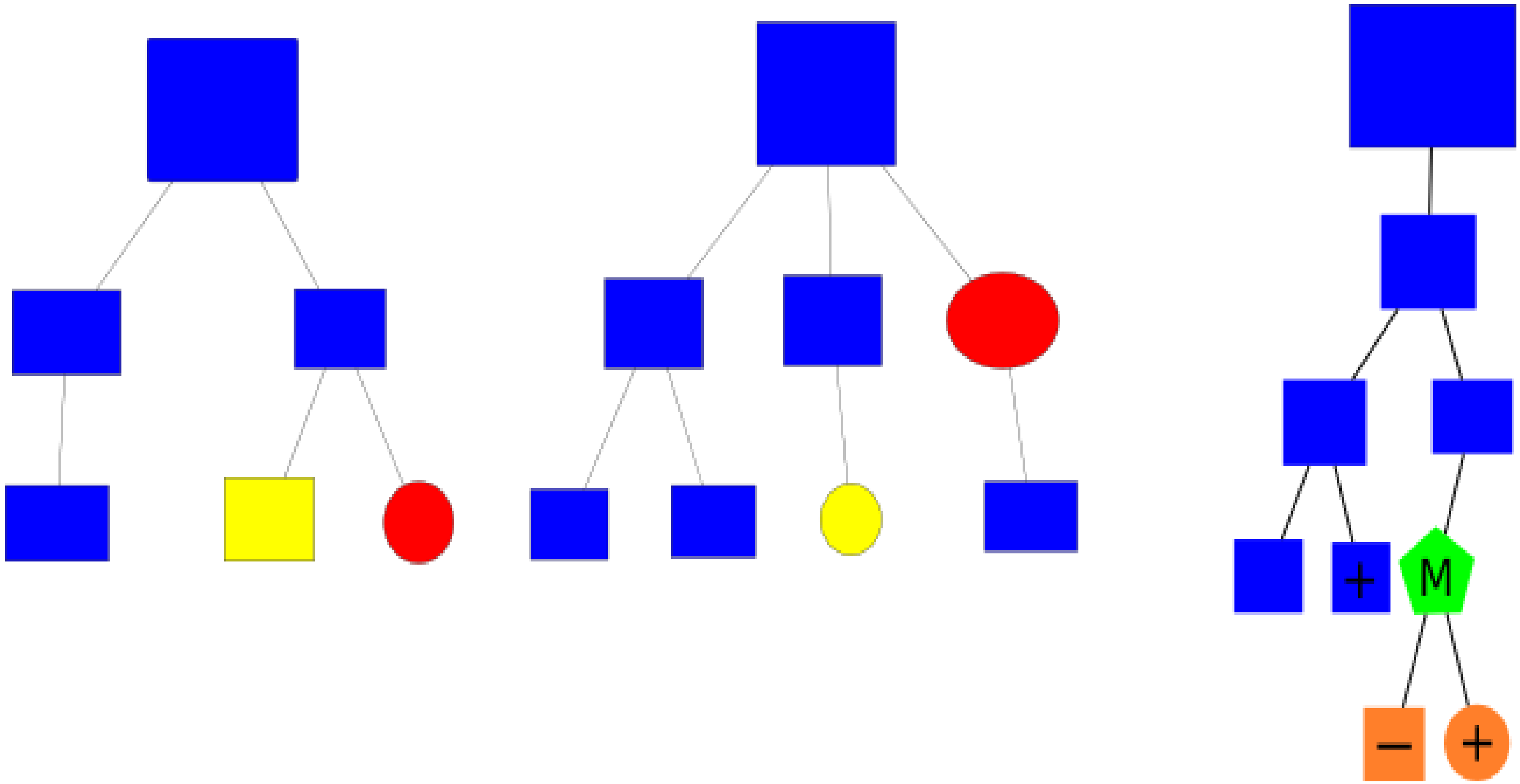
Algorithm (7)



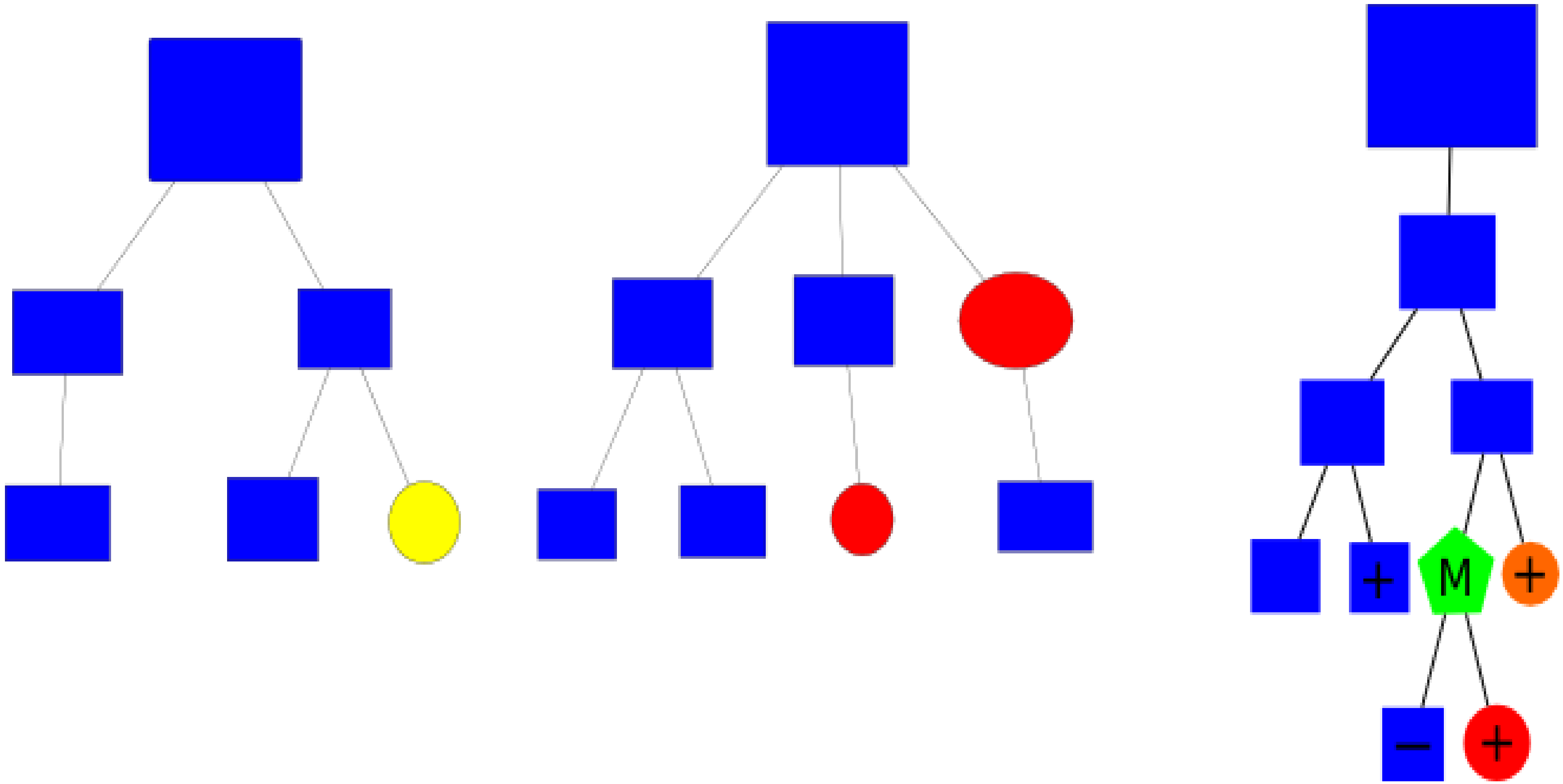
Algorithm (8)



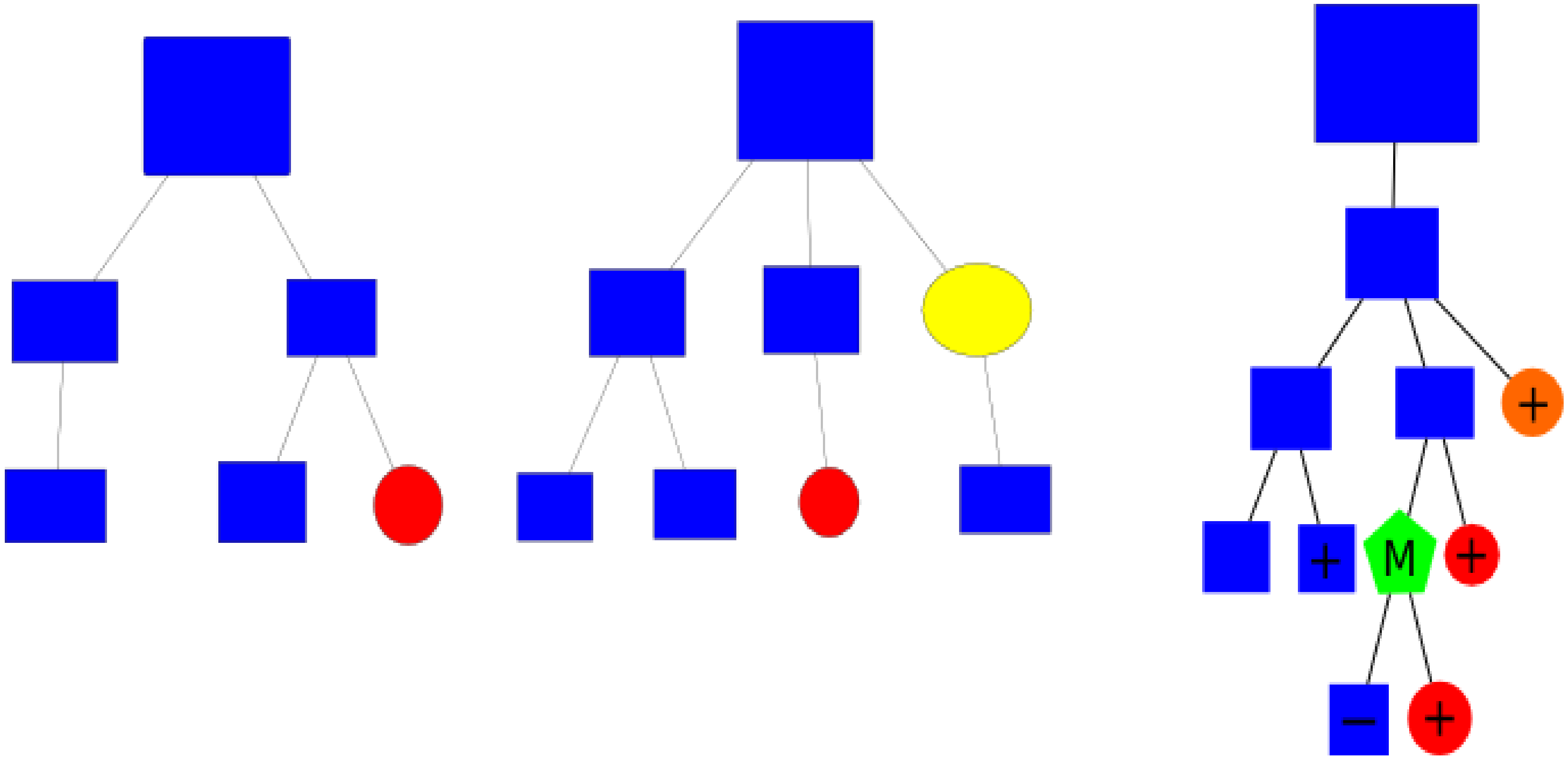
Algorithm (9)



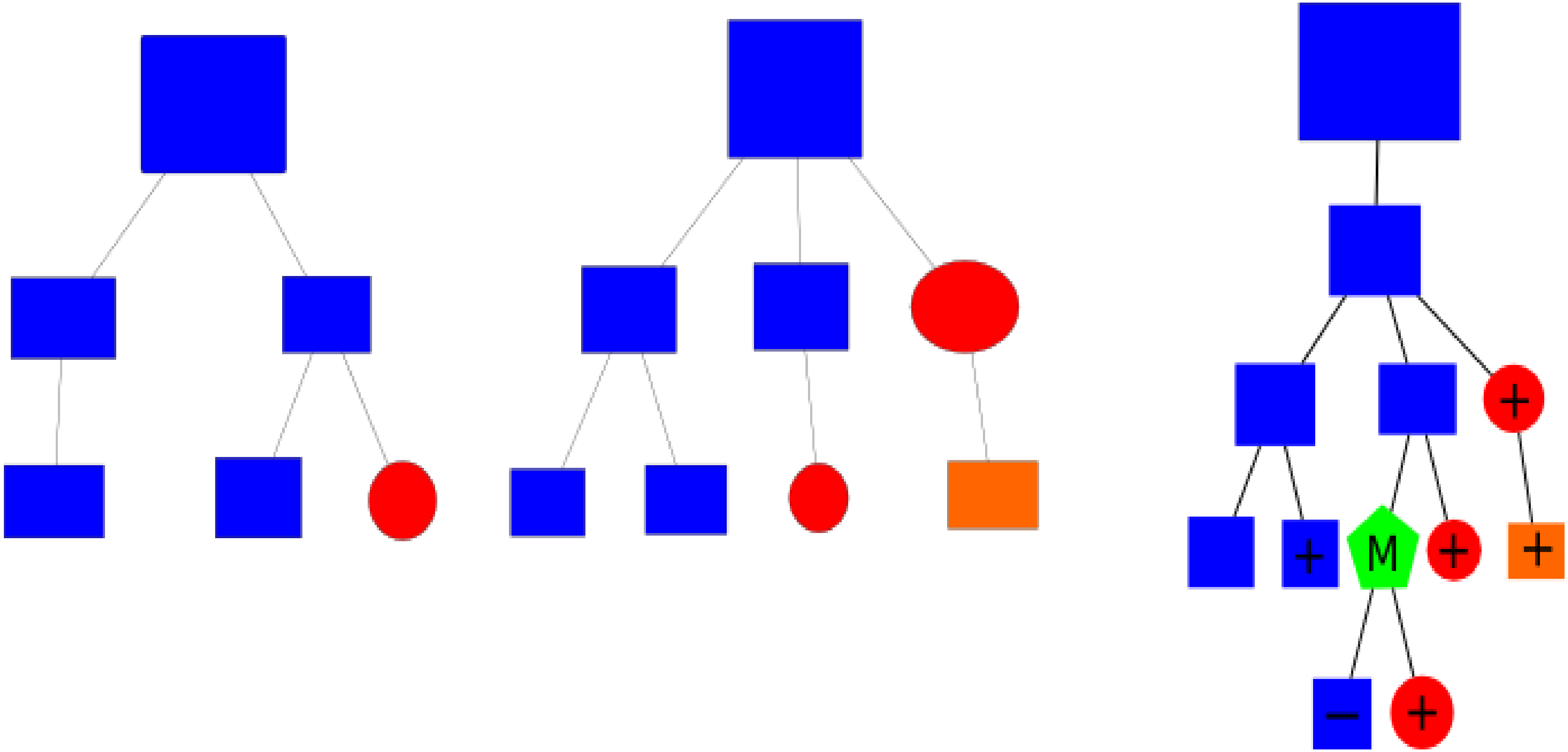
Algorithm (10)



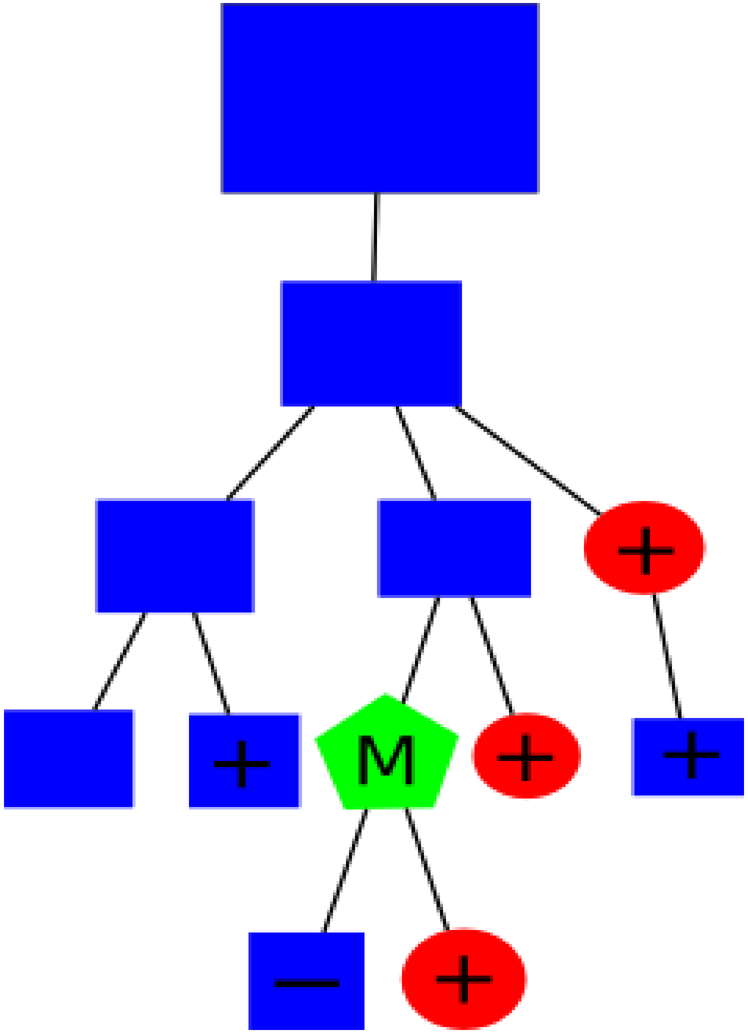
Algorithm (11)



Algorithm (12)



Algorithm (Final Output)



Applications

– Change tracking in office documents

– De-duplication of graphics, office, and news files

– Automation of system tasks

– Finding “new” items in a feed

Testing

- Tested on 4 Documents
  - 2 XML files obtained by decompressing OpenOffice.org ODT Files
  - 2 SVG Image Files
- Similar files with slight differences
- Our program successfully detected the differences in both file types and returned the output in a tree to the user

Future Work

- Use namespaces to distinguish “diff” attributes from application domain attributes
- Create library for integrating with applications
- Create an XML schema/DTD outlining the format so that diff files can be validated
- Integration with XML applications

Questions?