CS 6999 Semantic Web Techniques; 12 Nov 2002

Practice Exam

Family Name First Name Student ID Signature

1) Consider these XML elements for the 'pre(fix)' and 'post(fix)' application of a unary function, here factorial (!), to its argument, here a variable (x):

```
<pre>
  <fun> ! </fun>
  <arg> <var> x </var> </arg>
</pre>

<post>
  <arg> <var> x </var> </arg>
  <fun> ! </fun>
</post>
```

Complete the following XSLT template - by just filling in the six versions of "___" - for the (XML-to-XML) transformation of 'prefix' applications into 'postfix' applications:

```
<xsl:template match="pre">
  <_____>
    <___><xsl:value-of select="___"/></___>
    _________
  </_____>
</xsl:template>
```

Could this transformation be 'inverted' - mapping 'postfix' applications to 'prefix' applications - without information loss (write in only "yes" or "no" here)?

2) This is simplified RDF metadata about three fictitious people:

```
<rdf:RDF>
  <rdf:Description about="http://www.home4many/john">
    <name>John Smith</name>
    <spouse rdf:resource="http://www.home4many/mary"/>
  </rdf:Description>
  <rdf:Description about="http://www.home4many/mary">
    <name>Mary Smith</name>
    <spouse rdf:resource="http://www.home4many/john"/>
  </rdf:Description>
  <rdf:Description about="http://www.home4many/babs">
    <name>Barbara Smith</name>
    <father rdf:resource="http://www.home4many/john"/>
    <mother rdf:resource="http://www.home4many/mary"/>
  </rdf:Description>
</rdf:RDF>
```

Draw the directed labeled graph (DLG) that constitutes the RDF diagram of this XML element (use space below). Hint: URLs, going into ovals, and texts, going into rectangles, may be arbitrarily shortened, as long as they remain unique (e.g.: '.../john' or just 'john'; 'John S' or just 'JS').

3) Consider the following RuleML program.:

```
<rulebase>
  <imp>                                  <fact>
    <_head>                                <_head>
      <atom>                                 <atom>
        <_opr>                                 <_opr>
          <rel>p</rel>                           <rel>r</rel>
        </_opr>                                </_opr>
        <var>x</var>                           <ind>1</ind>
        <var>y</var>                         </atom>
      </atom>                              </_head>
    </_head>                             </fact>
    <_body>
      <and>                              <fact>
        <atom>                             <_head>
          <_opr>                             <atom>
            <rel>q</rel>                       <_opr>
          </_opr>                                <rel>q</rel>
          <var>y</var>                         </_opr>
        </atom>                                <ind>2</ind>
        <atom>                               </atom>
          <_opr>                           </_head>
            <rel>r</rel>e                </fact>
          </_opr>                      </rulebase>
          <var>x</var>
        </atom>
      </and>
    </_body>
  </imp>
```

Write the corresponding Prolog program by just completing the five versions of "___":

```
_________ :- ______, ______.        ______.

                                    ______.
```

What could be implied by the program (write in here using Prolog syntax)?

4) Using Prolog or any other logic notation, give a program that expresses that an ancestor is either (1) a parent or (2) a parent of an ancestor.

Add facts representing two of the parent relations in your own family or in a fictitious family, mentioning you or a fictitious person, one of their parents, and one of his or her parents. Show a query that asks for all of the known ancestors, and a proof using at least one occurrence of the rule (2).

Finally, consider whether or not the bottom-up inference system that we studied (j-DREW BU), using subsumption tests, could run into an infinite loop with this program.