Facoltà di ingegneria dell'informazione

Ingegneria della conoscenza 2009–10

Appello del 7 luglio 2010

Parte II			
Cognome		Laureando?	si ☐
Nome		Matricola	

4
6 pt.

```
<x:schema xmlns:x="http://www.w3.org/2001/XMLSchema">
  <x:element name="a">
    <x:simpleType><x:restriction base="x:integer"><x:minInclusive="-1"/>
    </x:restriction></x:simpleType>
  </x:element>
  <x:complexType name="A" mixed="true">
    <x:sequence>
      <x:sequence minCardinality="0" maxCardinality="unbounded">
        <x:element ref="a"/>
        <x:element name="b">
           <x:complexType>
              <x:attribute name="att" type="x:string" use="required"/>
           </x:complexType>
      </x:element>
     </x:sequence>
      <xsd:choice minCardinality="0">
        <x:element name="a" type="x:string"/>
        <x:element name="b" type="x:string"/>
      </xsd:choice>
    </sequence>
  </x:complexType>
  <x:element name="doc" type="A"/>
</x:schema>
```

- Scrivere il più piccolo documento XML valido di tipo "doc".
- Descrivere per quanto possibile lo stesso linguaggio tramite un DTD, indicando i punti in cui intervengono approssimazioni.
- Implementare un programma Java che assumendo il documento valido rispetto all'XSD, stampa a video se il costrutto "choice" ha generato un elemento "b".

(continuare sul 4° foglio)

1 -
visto che tutte le sequence interne hanno cardinalità minima 0 il socumento minimale è:
<doc></doc>
essendo mixed anche la soluzione <doc>testo</doc> è corretta

2 -

```
<!ELEMENT doc (a|b|#PCDATA)*>
<!ELEMENT a #PCDATA>
<!ELEMENT b #PCDATA>
<!ATTLIST b
   att  CDATA #IMPLIED>
```

approssimazioni:

- visto che il contenuto è mixed non posso descrivere le combinazioni lecite di a e b
- non posso esprimere il vincolo >= -1 del contenuto delle "a" del primo tipo
- visto che non posso distinguere le b del primo e secondo tipo l'attributo att è opzionale

3 -

ad esempio mi basta controllare che ci sia un b senza l'attributo “att”

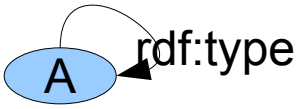
```
public class MyHandler
    extends DefaultnHandler
{
  boolean trovato;
  public void startDocument() {trovato = false;}

  public void startElement(String namespaceURI,String localName, String qName,  Attributes atts)
  {
     if (localName.equals(“b”) && atts.getValue(“att”) == null)
        trovato = true;
  }
  public void endDocument()
  {
     System.out.prinln(trovato);
   }
}
```

5 8 pt.	1 – descrivere le differenze tra i tre sottolinguaggi di OWL 1 2 – mostrare un modello (ABbox, TBox o entrambi) RDF che non è OWL DL 3 – per ogni frase, indicare l'espressività richiesta (RDF, OWL Lite, OWL DL) e, se possibile, rappresentarle graficamente 1. le persone possono essere fan di sportivi (che sono personaggi famosi) 2. le persone posso essere fan degli sportivi che sono famosi 3. le persone posso praticare degli sport, gli sportivi estremi praticano almeno uno sport estremo 4. le persone possono praticare degli sport gli spericolati praticano almeno due sport estremi

1 -

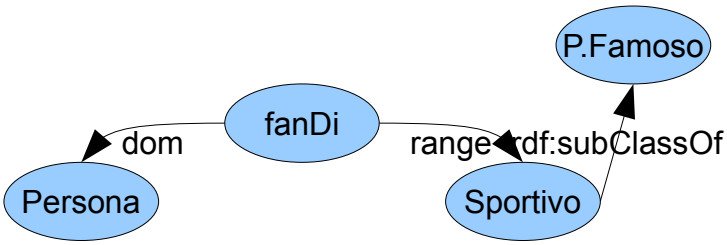
vedi dispense OWL-Lite / DL / Full

2 -

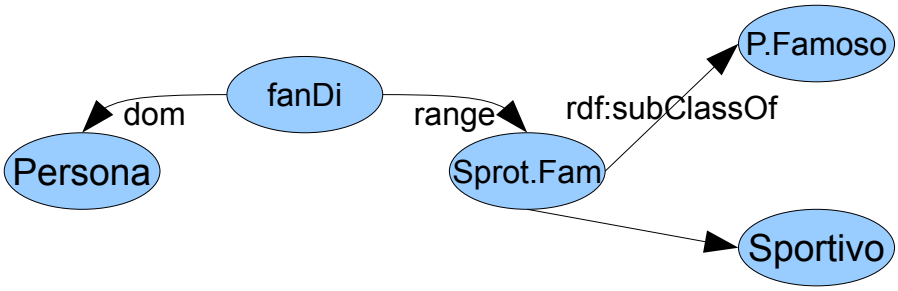


3 -

3.1 RDF

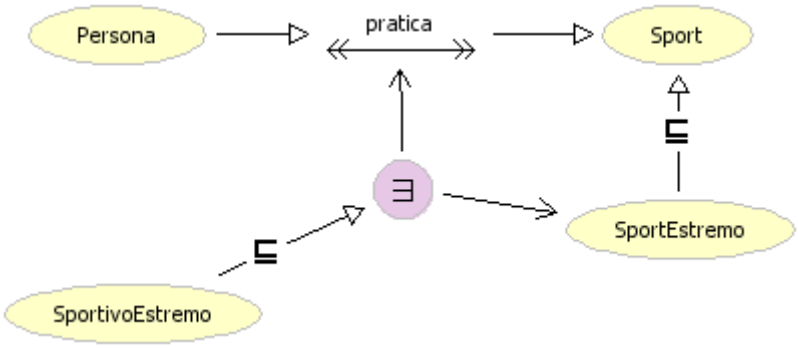


3.2 RDF



(o anche OWL-DL con l'intersezione tra P.Famoso e Sportivo al posto di Sport.Fam)

3.3 OWL-Lite (accettato anche DL)



3.4 nessuno dei tre (OWL 1 non supporta lo schema Q)

Allegati

DOM

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder builder = factory.newDocumentBuilder();
Document doc = builder.parse("libro.xml");
```

```
Interface Document{…
        Element getDocumentElement();
}
Interface Element extends Node {…
        String getAttribute(String name) ;
        public Node[ ] getChildNodes();
        String getTagName() ;
}
Interface CharacterData extends Node{…
        String getData();
}
```

SAX

```
Interface ContentHandler {…
        void startDocument();
        void startElement(String namespaceURI,String localName, String qName,  Attributes atts);
        void characters(char[] ch, int start, int length);
        void endDocument();
        void endElement(String namespaceURI,String localName, String qName);
}
Class DefaultHandler Implements ContentHandler();
interface Attributes{…
String getValue(String qName);
}
```