

MIF 04 GDW - TD 1

Correction

Exercice 1:

Pour chaque élément du document XML ci-dessous, donner son espace de nommage.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<categorie++ xmlns="http://macollection.com">
  <nom++ >
    Categorie <em** xmlns="http://www.w3.org/1999/xhtml">A</em></nom>
  <categorie>++
    <nom++ >B</nom>
    <categorie++ xmlns:xhtml="http://www.w3.org/1999/xhtml">
      <nom++ ><xhtml:em** >C</xhtml:em></nom>
    </categorie>
  </categorie>
  <coll:categorie++ xmlns="http://www.w3.org/1999/xhtml"
    xmlns:coll="http://macollection.com">
    <coll:nom++ ><b** >D</b></coll:nom>
    <categorie** >
      <coll:nom++ >E</coll:nom>
    </categorie>
  </coll:categorie>
</categorie>
```

Correction: ++ ↔ "http://macollection.com", ** ↔ "http://www.w3.org/1999/xhtml"

Exercice 2: Schemas

On considère la DTD suivante :

```
1 <!DOCTYPE inscriptions [
2 <!ELEMENT inscriptions (ue+,etudiant*)>
3 <!ELEMENT ue (titre, resume, inscrit*)>
4 <!ATTLIST ue code ID #REQUIRED
5           niveau CDATA #REQUIRED>
6 <!ELEMENT titre (#PCDATA)>
7 <!ELEMENT resume (#PCDATA)>
8 <!ELEMENT inscrit EMPTY>
9 <!ATTLIST inscrit num IDREF #REQUIRED
10           semestre CDATA #REQUIRED>
11 <!ELEMENT etudiant (nom,adresse)>
12 <!ATTLIST etudiant num ID #REQUIRED>
13 <!ELEMENT nom (#PCDATA)>
14 <!ELEMENT adresse (#PCDATA)>
15 ]>
```

On considère également le schéma XML suivant :

```

1 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
2
3   <xsd:element name="inscriptions">
4     <xsd:complexType>
5       <xsd:sequence>
6         <xsd:element name="ue" maxOccurs="unbounded"
7           type="ueContent"/>
8         <xsd:element name="etudiant" maxOccurs="unbounded"
9           minOccurs="0" type="etudiantContent"/>
10      </xsd:sequence>
11    </xsd:complexType>
12  </xsd:element>
13
14  <xsd:complexType name="ueContent">
15    <xsd:sequence>
16      <xsd:element name="titre" type="xsd:string"/>
17      <xsd:element name="resume" type="xsd:string"/>
18    </xsd:sequence>
19    <xsd:attribute name="code" type="xsd:ID" use="required"/>
20    <xsd:attribute name="niveau" type="xsd:string" use="required"/>
21  </xsd:complexType>
22
23  <xsd:complexType name="etudiantContent">
24    <xsd:sequence>
25      <xsd:element name="nom" type="xsd:string"/>
26      <xsd:element name="adresse" type="xsd:string"/>
27      <xsd:element name="inscription" minOccurs="0"
28        maxOccurs="unbounded">
29        <xsd:complexType>
30          <xsd:attribute name="codeUe" type="xsd:IDREF"
31            use="required"/>
32          <xsd:attribute name="semestre" type="xsd:string"
33            use="required"/>
34        </xsd:complexType>
35      </xsd:element>
36    </xsd:sequence>
37    <xsd:attribute name="num" type="xsd:ID" use="required"/>
38  </xsd:complexType>
39
40 </xsd:schema>

```

Soit D_{DTD} l'ensemble des documents valides par rapport à la DTD et D_{XSD} l'ensemble des documents valides par rapport au schéma XML.

1. Donner un exemple de document valide par rapport à D_{DTD} , puis un autre pour D_{XSD} .

Correction: D_{DTD} :

```

<?xml version="1.0"?>
<!DOCTYPE inscriptions SYSTEM "inscriptions.dtd">
<inscriptions>
  <ue code="BDAV" niveau="4">
    <titre>Bases de donnees avancees</titre>

```

```

    <resume>
      XML, Datalog, Securite
    </resume>
    <inscrit num="123456" semestre="2010/Automne"/>
  </ue>
  <etudiant num="123456">
    <nom>Toto</nom>
    <adresse>Villeurbanne</adresse>
  </etudiant>
</inscriptions>

```

D_{XSD} :

```

<?xml version="1.0"?>
<inscriptions>
  <ue code="BDAV" niveau="4">
    <titre>Bases de donnees avancees</titre>
    <resume>
      XML, Datalog, Securite
    </resume>
  </ue>
  <etudiant num="123456">
    <nom>Toto</nom>
    <adresse>Villeurbanne</adresse>
    <inscription codeUe="BDAV" semestre="2010/Automne"/>
  </etudiant>
</inscriptions>

```

2. Donner une représentation de la DTD et du XML Schema sous forme de grammaires d'arbre régulières.

Correction:

DTD:

```

I → inscriptions () ( $U^+, E^*$ )
U → ue ((code,id,1),(niveau,string,1)) ( $T, R, ER^*$ )
ER → inscrit ((num,idref,1),(semestre,string,1)) ()
E → etudiant ((num,id,1)) ( $N, A$ )
T → titre () ( $St$ )
R → resume () ( $St$ )
N → nom () ( $St$ )
A → adresse () ( $St$ )
St → string

```

XSD:

```

I → inscriptions () ( $U^+, E^*$ )
U → ue ((code,id,1),(niveau,string,1)) ( $T, R$ )
E → etudiant ((num,id,1)) ( $N, A, UR^*$ )
UR → inscription ((codeUe,idref,1),(semestre,string,1)) ()
T → titre () ( $St$ )
R → resume () ( $St$ )
N → nom () ( $St$ )
A → adresse () ( $St$ )
St → string

```

3. Caractériser D_{DTD} et D_{XSD} : Ces deux ensembles sont ils disjoints?

Correction: non: un document sans inscription est valide par rapport aux deux schemas
L'un des deux ensembles contient-il l'autre?

Correction: non: s'i y a une inscription, elle n'est pas valable dans l'autre schéma
Ces deux ensembles sont-ils égaux?

Correction: non, sinon ils se contiendraient mutuellement

4. Donner une requête XQuery permettant de passer d'un document de D_{DTD} à un document de D_{XSD} .

Correction:

```
<inscriptions>
  {
    for $ue in /inscriptions/ue
    return
      <ue>{$ue/@*}
      {$ue/titre}
      {$ue/resume}
      </ue>
  }
  {
    for $etu in /inscriptions/etudiant
    return
      <etudiant>{$etu/@*}
      {$etu/*}
      {
        for $i in //inscrit[@num=$etu/@num]
        return
          <inscription codeUe="{ $i/./@code}">{$i/@semestre}</inscription>
      }
      </etudiant>
  }
</inscriptions>
```

5. Y a-t-il perte d'information?

Correction: Non, car il est possible d'écrire un programme XQuery faisant la transformation inverse et indentique (à l'ordre entre inscriptions près).

6. Donner un schéma relationnel permettant de stocker l'information contenue dans un fichier de D_{DTD} .

Correction:

```
CREATE TABLE UE(code VARCHAR(255) PRIMARY KEY,
                 niveau VARCHAR(255),
                 titre VARCHAR(255),
                 resume TEXT);
CREATE TABLE ETUDIANT(num INTEGER PRIMARY KEY,
                       nom VARCHAR(255),
                       adresse TEXT);
CREATE TABLE INSCRIPTION(num INTEGER REFERENCES ETUDIANT(num),
                           code VARCHAR(255) REFERENCES UE(code),
                           semestre VARCHAR(255),
                           PRIMARY KEY (num,code,semestre));
```