

Interrogation écrite

Exercice 1 : Lecture sax d'un fichier xml

On considère le gestionnaire **sax** défini par la classe **InterroHandler** suivante :

```
1 import org.xml.sax.* ;
2 import org.xml.sax.helpers.* ;
3
4 public class InterroHandler extends DefaultHandler {
5     private int n ;
6     private boolean fermante ;
7
8     public void startDocument() {this.n = 0 ; this.fermante = false ;}
9
10    public void endDocument() {System.out.println() ;}
11
12    public void startElement(String namespaceURI,
13                             String localName,
14                             String rawName,
15                             Attributes attributs) {
16        if (this.fermante) System.out.print(" , ") ;
17        System.out.print(localName + "-" + this.n + "( ") ;
18        this.n++ ;
19        this.fermante = false ;
20    }
21
22    public void endElement( String namespaceURI,
23                           String localName,
24                           String rawName) {
25        System.out.print(" )") ;
26        this.n++ ;
27        this.fermante = true ;
28    }
29 }
```

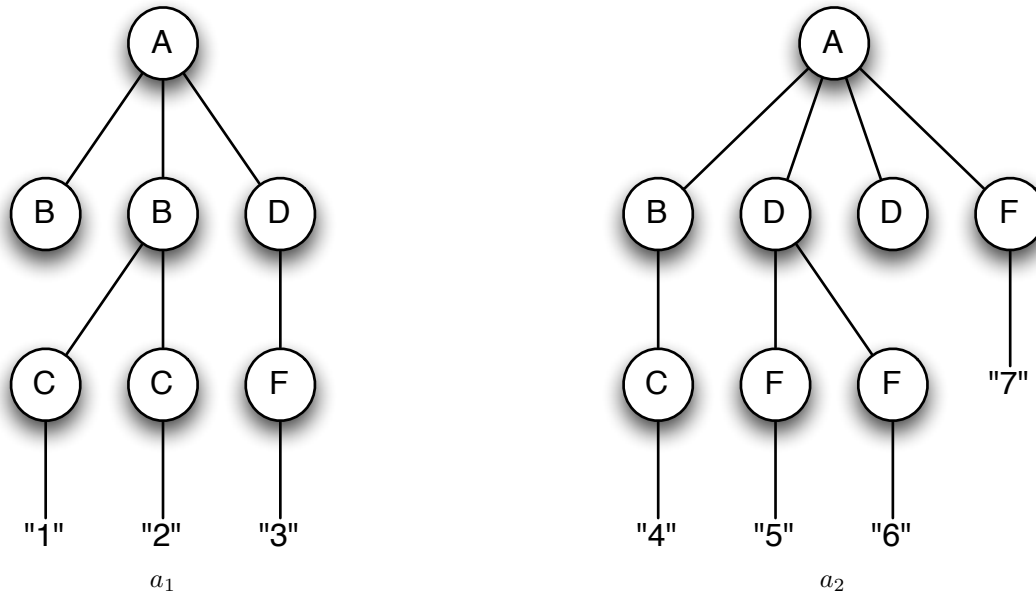
On considère aussi le fichier **a.xml** suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<a> <b/> <c> <d/> <e> <f/> </e> </c> <g/> </a>
```

Question 1 : Donner l'affichage obtenu en analysant le fichier **a.xml** à l'aide d'un gestionnaire **InterroHandler**.

Exercice 2 : DTD

On considère les deux arbres xml suivants a_1 et a_2 :



Question 1 : Traduire ces arbres en documents xml de noms respectifs `a1.xml` et `a2.xml`.

Question 2 : Définir une DTD qui valide le document `a1.xml` mais pas le document `a2.xml`.

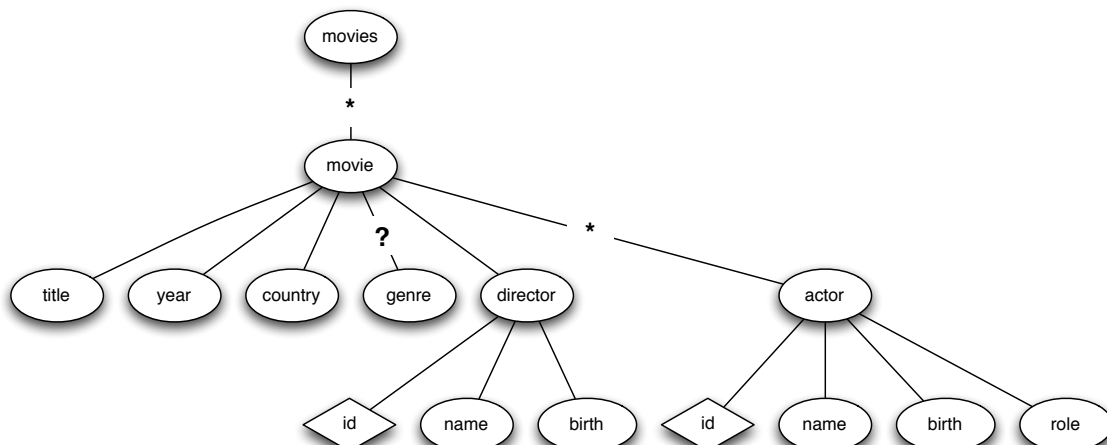
Question 3 : Définir une DTD qui valide le document `a2.xml` mais pas le document `a1.xml`.

Question 4 : Définir une DTD qui valide les deux documents `a1.xml` et `a2.xml`.

Exercice 3 : XPath

On considère des fichiers xml de descriptions de films dont le schéma est donné sous forme graphique où :

- un losange représente un attribut,
- un arc simple veut dire exactement 1 occurrence du sous-élément ou de l'attribut,
- un arc avec un point d'interrogation veut dire que le sous-élément (ou l'attribut) est optionnel
- un arc avec une étoile veut dire 0,1 ou plusieurs occurrences du sous-élément,



Voici un exemple d'instance valide pour ce schéma :

```
<?xml version="1.0" encoding="UTF-8"?>
<movies xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="movies.xsd">
  <movie>
    <title>Match Point</title>
    <year>2005</year>
    <country>USA</country>
    <genre>Crime</genre>
    <director id="14">
      <name>Woody Allen</name>
      <birth>1935</birth>
    </director>
    <actor id="15">
      <name>Jonathan Rhys Meyers</name>
      <birth>1977</birth>
      <role>Chris Wilton</role>
    </actor>
    <actor id="16">
      <name>Scarlett Johansson</name>
      <birth>1984</birth>
      <role>Nola Rice</role>
    </actor>
  </movie>
</movies>
```

Question 1 : D'après la balise ouvrante de l'élément **movies** de l'instance **xml** ci-dessus, le schéma est donné sous la forme d'un **XML-Schema** décrit dans un fichier de nom **movies.xsd**. Expliquer pourquoi ce schéma pourrait être aussi décrit par une DTD.

Question 2 : Que retourne la requête `count(//actor[@id="15"]/name/text())` si celle-ci est évaluée sur l'instance **xml** ci-dessus ?

Même question avec la requête `count(//actor[//@id="15"]/name/text())`

Question 3 : Écrire les requêtes **XPath** permettant d'extraire les informations suivantes :

1. Les titres des films dans lesquels joue **Scarlett Johansson**.
2. Les rôles joués par **Michel Blanc**.
3. Le nom du premier acteur dans la distribution de chaque film.
4. Les noms des réalisateurs des films dans lesquels jouent à la fois **Michel Blanc** et **Thierry Lhermite**.
5. Les titres des films dans lesquels le réalisateur joue aussi un rôle comme acteur.
6. Tous les noms d'acteurs en ne faisant figurer chaque nom qu'une seule fois.
7. Le nombre total d'acteurs du fichier interrogé.
8. Les noms des réalisateurs qui n'ont fait qu'un seul film.

Exercice 4 : XML-Schema

On considère le fichier `xml` suivant :

```
<?xml version="1.0" encoding="utf-8"?>

<societe xmlns="http://exo-schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://exo-schema
    http://www.fil.univ-lille1.fr/societe.xsd">
  <nom>CASTOR</nom>
  <adresse>
    <rue>2 rue du bateau</rue>
    <CP>59650</CP>
    <ville>Villeneuve d'Ascq</ville>
    <pays>France</pays>
  </adresse>
  <employe id="100">
    <nom>Durand</nom>
    <prenom>Paul</prenom>
    <mail>paul.durand@castor.com</mail>
    <statut>Ingenieur</statut>
  </employe>
  <employe id="59864">
    <nom>Dupond</nom>
    <prenom>Jacques</prenom>
    <mail>jacques.dupond@castor.com</mail>
    <statut>Agent-de-maitrise</statut>
  </employe>
  ...
  <services>
    <service responsable="100">
      <nom>Informatique</nom>
      <batiment>B12</batiment>
    </service>
    ...
  </services>
</societe>
```

Question 1 : Donner le (un) contenu (cohérent) de la balise ouvrante `schema`, racine du document `societe.xsd`.

Question 2 : Donner la définition du type de l'élément `statut` du schéma `societe.xsd`, sachant que les seules valeurs autorisées pour cet élément sont `Ingenieur` ou `Agent-de-maitrise`.

Question 3 : Donner la définition du type de l'attribut `id` du schéma `societe.xsd`, sachant qu'un `id` est une valeur entière comprise entre 100 et 99999.

Question 4 : Donner la définition de l'élément `employe` du schéma `societe.xsd`, sachant que l'attribut `id` est obligatoire.

Question 5 : On souhaite définir comme clé primaire, pour tout employé, la valeur de son attribut `id`. En précisant dans quelle définition d'élément du fichier `societe.xsd` faire apparaître cette définition de clé, donner les instructions `xml-schema` correspondantes.

Question 6 : On souhaite ajouter comme contrainte que le responsable d'un service doit être un membre du personnel. En précisant dans quelle définition d'élément du fichier `societe.xsd` faire apparaître cette contrainte, donner les instructions `xml-schema` correspondantes.

Exercice 5 : Espaces de noms

On considère le fichier `xml` suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<exercice xmlns:pre="http://A">
  <pre:niveau xmlns:pre="http://D" xmlns="http://A">
    <out:garage xmlns:out="http://C">
      <pre:alcove/>
      <entree xmlns:pre="http://E">
        <pre:cuisine/>
      </entree>
    </out:garage>
  </pre:niveau>
  <def xmlns="http://B">
    <pre:figue>
      <levain/>
    </pre:figue>
  </def>
</exercice>
```

Question 1 : Donner l'espace de nom de chacun des éléments `exercice`, `niveau`, `garage`, `alcove`, `entree`, `cuisine`, `def`, `figue` et `levain`.