

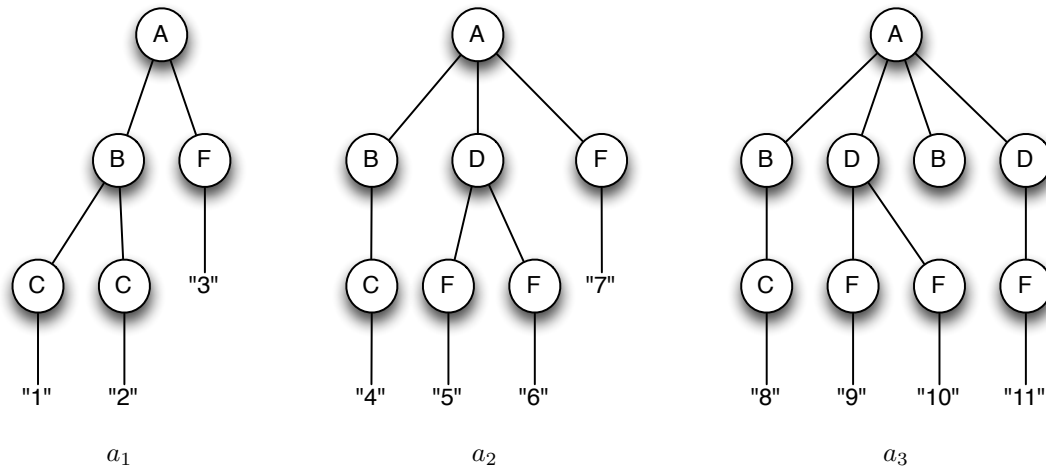
Traitement des données NoSQL

STI8 – EA Big Data, 2018–2019

Exercices à faire sur papier

Exercice 1

On considère les arbres XML suivants a_1 , a_2 , a_3 correspondant respectivement à des fichiers `a1.xml`, `a2.xml`, `a3.xml` :



On considère aussi les DTD suivants :

d1.dtd

```
<!ELEMENT A ((B|D)+,F)>
<!ELEMENT B (C*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (F*)>
<!ELEMENT F (#PCDATA)>
```

d2.dtd

```
<!ELEMENT A ((B,D)+,F?)>
<!ELEMENT B (C*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (F*)>
<!ELEMENT F (#PCDATA)>
```

d3.dtd

```
<!ELEMENT A (B+,D+,F*)>
<!ELEMENT B (C*)>
<!ELEMENT C (#PCDATA)>
<!ELEMENT D (F*)>
<!ELEMENT F (#PCDATA)>
```

1. Pour chacun de ces DTD, dire s'il valide ou non chacun des documents `a1.xml`, `a2.xml` et `a3.xml`.
2. Donner la forme sérialisée de chacun des documents `a1.xml`, `a2.xml` et `a3.xml`.

Exercice 2

Considérons le DTD suivant :

```
<!ELEMENT movies (movie*)>
<!ELEMENT movie (title, year, runtime, directors, actor*, summary)>
<!--ATTLIST movie id ID #REQUIRED-->
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT runtime (#PCDATA)>
<!ELEMENT directors (director*)>
<!ELEMENT director (name)>
<!ELEMENT actor (name, role)>
<!--ELEMENT name (firstname, lastname)-->
<!--ELEMENT firstname (#PCDATA)-->
<!--ELEMENT lastname (#PCDATA)-->
<!--ELEMENT role (#PCDATA)-->
<!--ELEMENT summary (#PCDATA)-->
```

1. Écrire un fichier XML valide par rapport à ce DTD.

2. Pour l'élément **name** comment changer la définition pour autoriser le nom et le prénom à apparaître dans n'importe quel ordre (mais exactement un nom et un prénom) ?
3. Même question pour **movie** (on veut autoriser les éléments à apparaître dans n'importe quel ordre, mais avec les même cardinalités que dans le DTD initial).

Exercice 3

Considérons le XML Schema suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="personnes">
    A COMPLETER
  </xs:element>
  <xs:element name="personne">
    A COMPLETER
  </xs:element>
  <xs:complexType name="type-evenement">
    A COMPLETER
  </xs:complexType>
</xs:schema>
```

Donner les définitions manquantes :

1. **personnes** : contient un nombre arbitraire d'éléments de type **personne**.
2. **personne** : contient (dans cet ordre) le nom, le prénom, la date et le lieu de naissance, et optionnellement la date et le lieu de décès. On retrouve également un attribut **id** obligatoire.
3. **type-evenement** : contient une date et un lieu (c'est le type à utiliser pour les naissances et les décès).

Par exemple, le document suivant doit être valide pour le XML schema résultat.

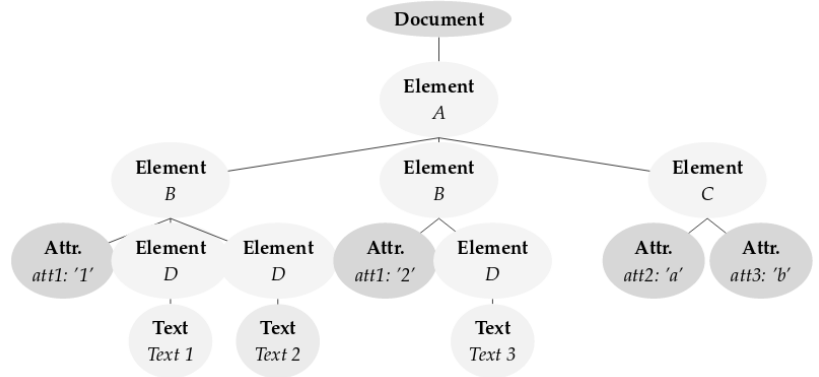
```
<?xml version="1.0" encoding="UTF-8"?>
<personnes>
  <personne id="p-034">
    <nom>Martin</nom>
    <prenom>Jules</prenom>
    <naissance>
      <date>1918-04-12</date>
      <lieu>Angers</lieu>
    </naissance>
    <deces>
      <date>1973-08-22</date>
      <lieu>Paris</lieu>
    </deces>
  </personne>
  <personne id="p-098">
    <nom>Martin</nom>
    <prenom>Henri</prenom>
    <naissance>
      <date>1931-08-14</date>
      <lieu>Paris</lieu>
    </naissance>
  </personne>
</personnes>
```

Exercice 4

Considérons l'arbre XML suivant.

Donnez les résultats des requêtes XPath suivantes.

1. `/descendant::B[1]/descendant::text()`
2. `/descendant::B[1]/descendant::text()[1]`
3. `/descendant::B[1]/descendant::text()[1]`
4. `/descendant::B/D/text()[1]`



Exercice 5

Expliquez la différence entre les requêtes suivantes. Puis, donnez un exemple pour lequel le résultat est identique et un exemple pour lequel on n'obtient pas le même résultat.

- `//livre[titre="edition"]`
- `//livre[titre=edition]`

Exercice 6

Expliquez la signification des requêtes suivantes. Lesquelles donneront toujours le même résultat ?

1. `/item/livre[@titre="bda" and position()=last()]`
2. `/item/livre[@titre="bda"] [position()=last()]`
3. `/item/livre[position()=last()] [@titre="bda"]`

Donnez un exemple pour lequel les trois requêtes ne donnent pas toutes le même résultat.

Exercice 7

On considère le fichier XML suivant, qui représente le stock d'un maraîcher. Donnez les requêtes XPath pour sélectionner les éléments suivants (ces requêtes doivent fonctionner sur tout document de même nature que l'exemple).

1. Les producteurs de fruits.
2. Les légumes produits en Espagne.
3. Les origines des clémentines issues de l'agriculture biologique.
4. Les producteurs bretons.

```

<?xml version="1.0" encoding="UTF-8"?>
<produits>
  <fruit type="clementine" prix="290" calibre="1">
    <producteur>Production Bastia</producteur>
    <origine region="Corse">France</origine>
    <qty>15</qty>
    <note>Sans pepins, avec feuilles</note>
    <bio/>
  </fruit>
  <fruit type="clementine" prix="290" calibre="1">
    <producteur>Production Bastia</producteur>
    <origine region="Corse">France</origine>
    <qty>15</qty>
    <note>Sans pepins, avec feuilles</note>
  </fruit>
  <legume type="courgette" prix="300" calibre="2">
    <producteur>Madrid Hortelano</producteur>
    <origine>Espagne</origine>
    <qty>100</qty>
    <bio/>
  </legume>
  <legume type="chou fleur" prix="090" calibre="2">
    <producteur>Pontivy et Cie</producteur>
    <origine region="Bretagne">France</origine>
    <qty>100</qty>
  </legume>
  <legume type="salade" prix="075" calibre="3">
    <producteur>Marius Production</producteur>
    <origine region="Provence">France</origine>
    <qty>35</qty>
    <note>Batavia</note>
  </legume>
  <fruit type="melon" prix="150" calibre="1">
    <producteur>Marius Production</producteur>
    <origine region="Provence">France</origine>
    <qty>50</qty>
    <note>Melon brode</note>
    <bio/>
  </fruit>
</produits>

```