

**Données du Web**  
**Partie Federico Ulliana - (10 points)**  
**Documents non autorisés**

Numéro Étudia

**La modélisation des données en XML.**

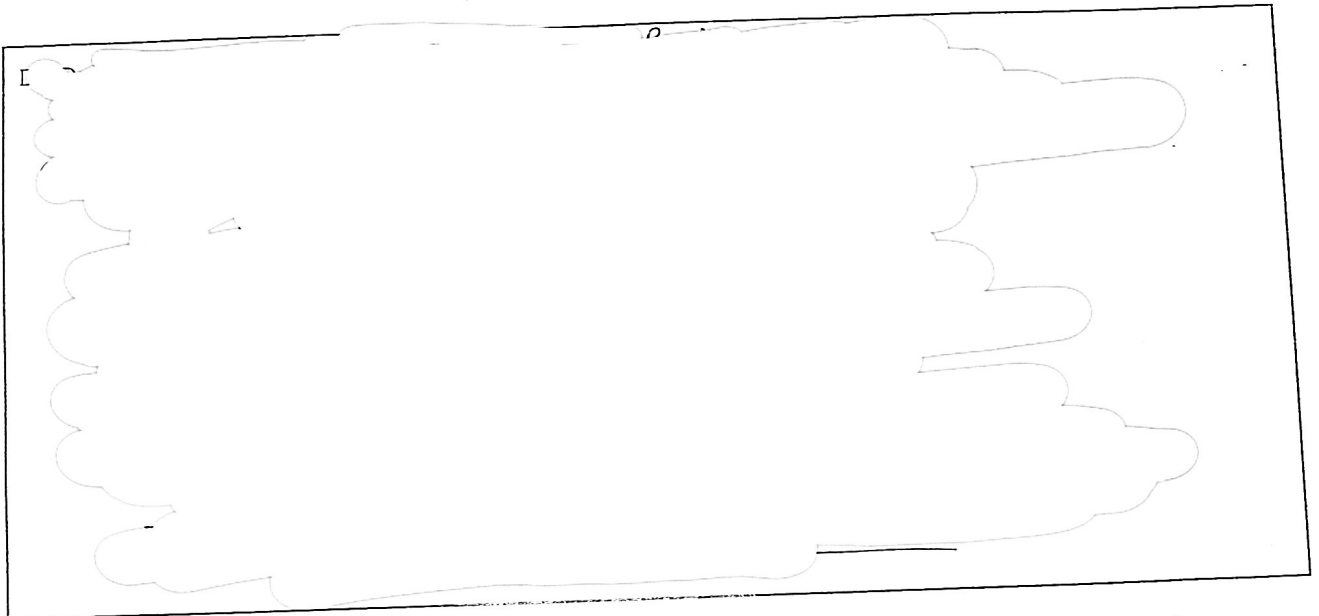
Internet Movie Database (IMDB) est une application Web sur le cinéma et la télévision mondiale. La plateforme met à disposition une API permettant à d'autres services Web de télécharger ses données. Ici, nous nous intéressons en particulier aux films, dont on peut récupérer (parmi d'autres) le titre, le genre, le metteur en scène, la liste des acteurs, l'année d'apparition, et la langue. Bien évidemment, certaines de ces informations peuvent avoir des identifiants associés.

L'exportation des données au format XML d'IMDB suit les spécifications d'une DTD définissant la structure des données exportées.<sup>1</sup>

**Question :**

1. Donner 2 DTD définissant deux structures possibles des données exportées.
  - a. La première DTD doit éviter la redondance des données.
  - b. La deuxième doit permettre l'évaluation efficace des requêtes du style  
*Q : dans quelles années Charlie Chaplin a joué une comédie ?*

2. Pour chaque DTD proposée, exprimer la requête *Q* en XPath.



<sup>1</sup>Attention : on ne s'intéresse pas à comment la plateforme stocke ses données.

Répondre aux questions dans l'énoncé.

## L'égalité dans XQuery

L'opérateur d'égalité du langage XQuery a une sémantique particulière conçue pour rendre son évaluation efficace. Ici, on souhaite aller au delà de son utilisation et définir une fonction XQuery permettant de tester si deux séquences de valeurs XQuery sont réellement identiques.

Pour simplicité, on se limitera aux séquences contenant seulement des *entiers* et des *arborescences* (*mais sans attributs ou texte*). Les arborescences sont indiquées par des noeuds de type élément. Par exemple,  $(5, \langle b \rangle \langle a \rangle \langle b \rangle)$  est une séquence contenant deux valeurs (séparées par une virgule) : une de type entier et une de type élément (correspondant à la racine de l'arborescence).

Question : définir une fonction XQuery `new-equality(Sx, Sy)` qui restitue `true` si les deux séquences liées aux variables `Sx` et `Sy` sont strictement identiques et `false` autrement.<sup>2</sup> Quelques exemple :

```
new-equality( (⟨b⟩⟨a⟩⟨b⟩, 5) , (⟨b⟩⟨a⟩⟨b⟩, 5) , )  
restitue true
```

```
new-equality( (⟨c/⟩) , (⟨c/⟩, ⟨c/⟩) , )  
restitue false
```

```
new-equality( (5) , ( ⟨c/⟩, 5) , )  
restitue false
```

Suggestions : vous pouvez utiliser le constructeur booléen `instance of` pour tester le type d'une valeur (l'utilisation est identique à celle du langage Java).

|                          |                                      |                       |                    |
|--------------------------|--------------------------------------|-----------------------|--------------------|
| <code>⟨b/⟩</code>        | <code>instance of element</code>     | <code>restitue</code> | <code>true</code>  |
| <code>⟨a⟩⟨b/⟩⟨/a⟩</code> | <code>instance of xsd:integer</code> | <code>restitue</code> | <code>false</code> |
| <code>5</code>           | <code>instance of xsd:integer</code> | <code>restitue</code> | <code>true</code>  |

XQuery met aussi à disposition une fonction `name(Sx)` permettant d'extraire la balise d'un noeud de type élément (le noeud étant lié à la variable `Sx`). Vous avez aussi le droit d'utiliser toutes les fonctions du langage qui peuvent vous aider.

<sup>2</sup>en XQuery `true` est noté `true()`, mais c'est un détail négligeable ici. Le même vaut pour `false`.

**Répondre aux questions dans l'énoncé.**

### Expressions régulières et déterminisme.

1. Dire si les expressions régulières suivantes sont déterministes.
2. Construire le graphe associé aux expressions.
3. Donner une séquence d'éléments valide et une séquence d'éléments non-valide pour les deux expressions.

$(b|(f,d,b))^*, (b|(d,e,b))^*, e$

REG1

$(b^*|(f|b))|(b,(b|c))$

REG2

Répondre aux questions dans l'énoncé.

Examen Données du web HMIN103

06/01/2020 - 8h30-10h30  
TD 36.402 (73), TD 36.415 (1)

Pierre Pompidor

Aucun document autorisé

1. Pourquoi les "tables" sous MongoDB sont-elles appelées des collections ? (0,5 points)
2. Sous MongoDB, quelle méthode permet de modifier la valeur d'une propriété d'un document ? (0.5 point)
3. Qu'est-ce un service web ? (1 point)
4. Comment les objets JavaScript insérés dans le corps d'une requête HTTP de méthode POST sont-ils manipulables dans un service web géré par Node.js ? (1 point)
5. Dans l'architecture MEAN, où est géré le "Modèle" du paradigme de conception MVC ? (1,5 point)
6. Dans une application Angular, à quels moments les instances des classes TypeScript sont-elles allouées ? (1,5 points)
7. Ecriture d'une méthode d'un composant Angular (2 points)

Ecrivez la méthode *ngOnInit()* de la classe *ProduitsComponent* qui value son attribut *produits* en utilisant la méthode *getProduits()* du service *ProduitsService*.

8. Ecriture d'un template (2 points)

Ecrivez le template associé à la classe *ProduitsComponent* qui affiche dans une liste HTML le nom et la marque des produits.

La valeur de la propriété *stock* sera également affichée à condition que l'observable *admin* contienne une valeur différente de chaîne vide.