

Exam Master Info Orsay - WDMD

3 mars 2009

Documents non autorisés. Durée: 2 heures. L'examen consiste en 3 questions - 1,2,3 - vous pouvez remplacer l'une d'entre elles par la question 4 pour 6 points

You can use English to answer. No document. The exam consists in 3 questions - 1,2,3 - you can substitute one of them with Question 4 for 6 points.

1 Requêtes XPath et XQuery (6 points)

Voici un document sur les films.

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

<movie>
  <title>Match Point</title>
  <year>2005</year>
  <country>USA</country>
  <genre>Crime</genre>
  <director id='14' >
    <last_name>Allen</last_name><first_name>Woody</first_name>
    <birth_date>1935</birth_date>
  </director>
  <actor id='15' >
    <first_name>Jonathan</first_name><last_name>Rhys Meyers</last_name>
    <birth_date>1977</birth_date><role>Chris Wilton</role>
  </actor>
  <actor id='16' >
    <first_name>Scarlett </first_name><last_name>Johansson</last_name>
    <birth_date>1984</birth_date><role>Nola Rice</role>
  </actor>
</movie>
```

Ecrivez les requêtes XPath suivantes :

1. Les titres des films où joue Scarlett Johansson.
2. Les rôles joués par Michel Blanc
3. Le premier acteur dans la distribution de chaque film.
4. Les réalisateurs des films où jouent à la fois Michel Blanc ET Thierry Lhermite.
5. Les titres des films où le directeur joue aussi un rôle comme acteur.

Donner les trois dernières requêtes en XQuery, avec clauses FLOWR.

Solution :

1. `/movie[actor[first_name='Scarlett' and last_name='Johanson']]/title`
2. `/movie/actor[first_name='Michel' and last_name='Blanc']/role`
3. `/movie/actor[1]`
4. `/movie[actor[first_name='Michel' and last_name='Blanc'] and actor[first_name='Thierry' and last_name='Lhermite']]/director`

5. `/movie[director/@id = actor/@id]/title`

2 Réursion en XQuery (8 points)

On s'intéresse aux documents MathML qui décrivent des formules mathématiques en notation préfixée. (les opérateurs apparaissent avant les opérandes). Voici la représentation de la formule $x^2 + 4x + 4$

`(+ (^ x 2) (* 4 x) 4)`

Et voici le document MathML.

```
<?xml version="1.0" encoding="UTF-8"?>
<apply>
  <plus/>
  <apply>
    <power/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <apply>
    <times/>
    <cn>4</cn>
    <ci>x</ci>
  </apply>
  <cn>4</cn>
</apply>
```

On suppose dans tout ce qui suit que les seuls opérateurs connus sont l'addition (`<plus/>`) et la multiplication (`<times/>`), et que ces opérateurs sont binaires (exactement deux opérandes).

1. Donner l'expression en XQuery qui transforme un élément `apply` sans récursion en forme infixe. Par exemple, appliquée au document suivant:

```
<apply>
  <times/>
  <ci>x</ci>
  <cn>2</cn>
</apply>
```

la requête renvoie la chaîne "x * 2".

Solution :

```
for $op in document("mathml/mathml_basic.xml")/*/*[self::plus or self::times]
return
  <result>
    {$op/following-sibling::*[1]/text()}
    {if (name($op) = "plus") then "+" else "*"}
    {$op/following-sibling::*[2]/text()}
  </result>
```

2. Supposons que l'on puisse nommer la requête précédente comme une fonction `eval($op, $x, $y)`, où `$op` est une opération, `$x` et `$y` deux opérandes. On peut appeler récursivement toute fonction. Donner la requête XQuery qui renvoie une expression MathML contenant des `apply` récursifs sous forme infix. Par exemple, appliquée au document suivant:

```
<apply>
  <times/>
  <apply>
    <plus/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <ci>y</ci>
</apply>
```

la requête renvoie la chaîne "(x + 2) * y".

Solution :

```
declare function local:eval($op, $x, $y) {
  let $dummyVariable := "aa"
  return
    concat(' ( ' ,
      if (name($x) = "apply")
      then local:eval($x/*[1], $x/*[2], $x/*[3])
      else $x/text(),
      if (name($op) = "plus") then "+" else "*",
      if (name($y) = "apply")
      then local:eval($y/*[1], $y/*[2], $y/*[3])
      else $y/text(),
      ') ' )
};

let $mydoc := doc("mathml/mathml.xml")
return
<resultContainer>{local:eval($mydoc/apply/*[1],
$mydoc/apply/*[2],
$mydoc/apply/*[3])}
</resultContainer>
```

3 Réseaux pair-à-pair (6 points)

Répondre en moins d'une page.

Des activistes politiques vivant en dictature cherchent à concevoir un système de *groupes de discussions* (ou *forum*) pour communiquer entre eux. Ce système doit avoir toutes les fonctionnalités classique d'un forum de discussion: les *utilisateurs* lisent et écrivent des *messages* se répondant les uns aux autres dans des *enfilades* (ou *threads*) regroupés dans des *groupes*.

Trois systèmes sont envisagés: un système centralisé, un système en pair-à-pair non structuré, et un système en pair-à-pair structuré s'appuyant sur une table de hachage distribuée.

1. Présenter rapidement, pour chacune de ces alternatives, où et comment les données seront stockées.
2. Indiquer les avantages et inconvénients (en proposant éventuellement des solutions à ces inconvénients) de chacune de ces 3 alternatives vis-à-vis des problèmes suivants:

- (a) efficacité (rapidité) générale du système;
- (b) résistance à la censure;
- (c) préservation de l'anonymat.

4 Automates d'arbres (Bonus)

Soit q la requête Booléenne exprimée en XPath par:

$$q_{17} = //a[./b_1]...[./b_{17}]$$

Elle vérifie si un arbre XML a un noeud étiqueté a qui a des descendants étiquetés b_1, \dots, b_{17} . On peut définir en général q_n .

Pour un n fixé, la requête q_n est-elle exprimable (i) par un automate d'arbre non-déterministe, (ii) par un automate d'arbre déterministe. Expliquez vos réponse en une ou deux phrases (courtes) en mentionnant s'il y a lieu, le nombre d'états de l'automate.

Solution :

On peut l'exprimer avec un automate d'arbre non-déterministe bottom-up. On remonte l'information "j'ai vu cet ensemble de b_i ".

Remarque: Donc, on peut aussi l'exprimer avec un top-down non-déterministe ou un bottom-up déterministe. On ne peut pas l'exprimer avec un top-down déterministe.