

Bases de Données II, Mons

Jef Wijsen

1 juin 2010

Cahier fermé. Durée : 3 heures

Nom et prénom
Année

La figure 2 montre une base de données XML concernant des peintures et des fresques. La base enregistre, entre autres, que la Mona Lisa est une peinture de Leonardo Da Vinci et que L'Ultima Cena (la Cène) est une fresque de Leonardo Da Vinci. Leonardo da Vinci, identifié par le code LV, est né en 1425 et mort en 1519. La figure 1 montre le DTD.

```
<!-- This file is called art.dtd -->
<!ELEMENT ART (ARTISTS, PAINTINGS, FRESCOS)>
<!ELEMENT ARTISTS (artist)*>
<!ELEMENT PAINTINGS (painting)*>
<!ELEMENT FRESCOS (fresco)*>
<!ELEMENT painting (title, artist)>
<!ELEMENT fresco (title, artist)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT artist (#PCDATA)>
<!ATTLIST artist id CDATA #REQUIRED>
<!ATTLIST artist sex CDATA #IMPLIED>
<!ATTLIST artist birth CDATA #IMPLIED>
<!ATTLIST artist death CDATA #IMPLIED>
<!ATTLIST painting year CDATA #REQUIRED>
<!ATTLIST fresco year CDATA #REQUIRED>
```

Figure 1: DTD.

```

<?xml version="1.0"?>
<!DOCTYPE ART SYSTEM "art.dtd">
<ART>
<ARTISTS>
  <artist id="PP" sex="M" birth="1881" death="1973">Pablo Picasso</artist>
  <artist id="LV" sex="M" birth="1452" death="1519">Leonardo Da Vinci</artist>
  <artist id="MA" sex="M" birth="1475" death="1564">Michelangelo</artist>
</ARTISTS>
<PAINTINGS>
  <painting year="1904"><title>The Madman</title>
    <artist id="PP"/>
  </painting>
  <painting year="1939"><title>Night Fishing at Antibes</title>
    <artist id="PP"/>
  </painting>
  <painting year="1507"><title>Mona Lisa</title>
    <artist id="LV"/>
  </painting>
  <painting year="1475"><title>The Annunciation</title>
    <artist id="LV"/>
  </painting>
  <painting year="1506"><title>Doni Tondo</title>
    <artist id="MA"/>
  </painting>
</PAINTINGS>
<FRESCOS>
  <fresco year="1498"><title>L'Ultima Cena</title>
    <artist id="LV"/>
  </fresco>
  <fresco year="1512"><title>Sistine Chapel ceiling</title>
    <artist id="MA"/>
  </fresco>
</FRESCOS>
</ART>

```

Figure 2: Fichier XML avec des informations sur des peintures et des fresques.

Question 1 Écrivez une expression XPath qui rend la date de décès de Leonardo Da Vinci.

.../4

Question 2 Écrivez une expression XPath qui rend le nom de l'artiste qui a peint la fresque *L'Ultima Cena*.

.../4

Question 3 Écrivez une expression XPath qui rend les titres des peintures de Pablo Picasso. Notez : l'expression doit utiliser le nom "Pablo Picasso", pas l'identifiant "PP". Voici le résultat :

```
<title>The Madman</title>  
<title>Night Fishing at Antibes</title>
```

.../4

Question 4 Écrivez un programme XSLT qui affiche pour chaque artiste les peintures et les fresques qu'il a peintes, dans le format qui suit :

```
<?xml version="1.0" ?>
<artistes>
  <artiste><nom>Pablo Picasso</nom>
    <peintures><titre>The Madman</titre>
      <titre>Night Fishing at Antibes</titre>
    </peintures>
    <fresques />
  </artiste>
  <artiste><nom>Leonardo Da Vinci</nom>
    <peintures><titre>Mona Lisa</titre>
      <titre>The Annunciation</titre>
    </peintures>
    <fresques><titre>L'Ultima Cena</titre>
    </fresques>
  </artiste>
  <artiste><nom>Michelangelo</nom>
    <peintures><titre>Doni Tondo</titre>
    </peintures>
    <fresques><titre>Sistine Chapel ceiling</titre>
    </fresques>
  </artiste>
</artistes>
```

.../10

Question 5 A, B, C, \dots sont des entreprises d'autobus (le nombre d'entreprises n'est pas connu a priori). La table *Trajet* donne les trajets assurés par chacune des entreprises. On dit que l'entreprise X a le monopole du trajet ville1–ville2 si les deux conditions suivantes sont vérifiées :

1. $Trajet(\text{ville1}, \text{ville2}, X)$ est vrai; et
2. si X fera faillite, il ne sera plus possible d'atteindre ville2 à partir de ville1.

Par exemple,

<i>Trajet</i>	Départ	Arrivée	Entreprise
	Mons	Charleroi	A
	Charleroi	Bruxelles	A
	Bruxelles	Mons	A
	Bruxelles	Charleroi	B
	Charleroi	Bruxelles	B
	Bruxelles	Ath	C
	Ath	Mons	C
	Mons	Bruxelles	C

Notez que l'entreprise A n'a pas le monopole du trajet Mons–Charleroi, parce que même si A cesse ses activités, on peut aller de Mons à Charleroi via $Trajet(\text{Mons}, \text{Bruxelles}, C)$ suivi de $Trajet(\text{Bruxelles}, \text{Charleroi}, B)$. Pour cette table, il n'y a que deux monopoles : C a le monopole des trajets Bruxelles–Ath et Ath–Mons.

Donnez un programme datalog stratifié qui calcule un prédicat ternaire *Monopole*, tel que $Monopole(v, w, x)$ est vrai si x est une entreprise qui a le monopole du trajet $v-w$. **Expliquez en français l'idée derrière votre programme.**

... /10

Question 6 Soit q la requête conjonctive suivante :

$$q : Ans(x, y) \leftarrow E(x, u), E(z, u), E(z, v), E(y, v)$$

Soit P le programme datalog suivant :

$$P : \begin{cases} Ans(u, v) \leftarrow E(u, x), E(v, x) \\ Ans(u, v) \leftarrow Ans(u, w), E(w, x), E(v, x) \end{cases}$$

Indiquez si les expressions suivantes sont vraies ou fausses (cochez deux cases).

	Vrai	Faux
$q \sqsubseteq P$	<input type="checkbox"/>	<input type="checkbox"/>
$P \sqsubseteq q$	<input type="checkbox"/>	<input type="checkbox"/>

Argumentez pourquoi l'expression $q \sqsubseteq P$ est Vraie/Fausse.

<input type="text" value=".../5"/>

Argumentez pourquoi l'expression $P \sqsubseteq q$ est Vraie/Fausse.

<input type="text" value=".../5"/>

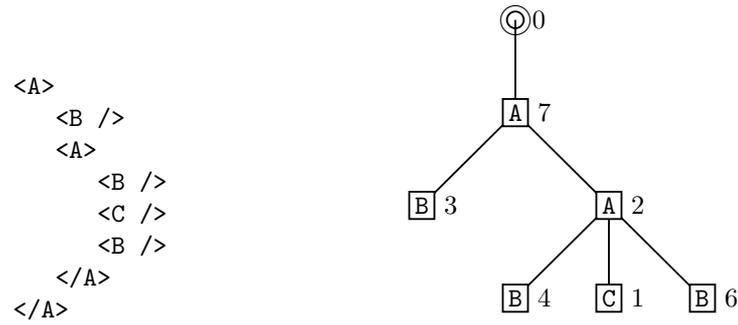


Figure 3: Document et arbre XML.

Question 7 Dans cet exercice, on encodera en datalog des documents XML et les axes XPath. Pour des raisons de simplicité, on ne considère que deux types de nœud : *the root node* et *element nodes*. Chaque nœud est identifié par un nombre naturel. La numérotation ne suit aucune logique; la seule contrainte est que deux nœuds différents ne peuvent pas porter le même nombre. L'encodage de l'arbre en datalog utilise quatre prédicats *edb*, à savoir *Root*, *Child*, *NextSibling*, *Label*.

- *Root*(i) est vrai si le nœud i est la racine (*the root node*);
- *Child*(i, j) est vrai si le nœud j est un enfant du nœud i ;
- *NextSibling*(i, j) est vrai si j est un *sibling* de i qui vient juste après i ;
- *Label*(i, X) est vrai si i est un *element node* de type X .

La figure 3 montre un document XML (à gauche) et son arbre (à droite) dans lequel les nœuds sont numérotés. Cet arbre sera encodé par la base de données suivante :

$$\{ \text{Root}(0), \text{Child}(0, 7), \text{Child}(7, 3), \text{Child}(7, 2), \text{Child}(2, 4), \text{Child}(2, 1), \text{Child}(2, 6), \text{NextSibling}(3, 2), \text{NextSibling}(4, 1), \text{NextSibling}(1, 6), \text{Label}(1, C), \text{Label}(2, A), \text{Label}(3, B), \text{Label}(4, B), \text{Label}(6, B), \text{Label}(7, A) \}$$

Écrivez un programme datalog pour le prédicat *idb Following*, défini comme suit :

Following(i, j) \iff si i est le nœud de contexte, alors le nœud j est dans l'axe **following**, suivant la sémantique d'XPath.

Par exemple, *Following*(3, 2), *Following*(3, 4), *Following*(3, 1), *Following*(3, 6), *Following*(4, 1), *Following*(4, 6), *Following*(1, 6). **Expliquez en français l'idée derrière votre programme.**

.../10

Question 8 Voici un programme datalog⁷.

$$\begin{aligned} S(x) &\leftarrow P(x, x), \neg R(x, x) \\ R(x, y) &\leftarrow P(x, y), C(x, "r") \\ R(x, y) &\leftarrow P(x, y), C(y, "r") \\ P(x, y) &\leftarrow F(x, y) \\ P(x, z) &\leftarrow P(x, y), F(y, z) \\ A(x) &\leftarrow P(x, x), \neg S(x) \end{aligned}$$

Détaillez (montrez les stratum, l'ordre des règles, les points fixes, etc.) comment ce programme est exécuté sur la base de données suivante :

F	$\begin{array}{ c c } \hline 1 & 2 \\ \hline 1 & 2 \\ \hline 2 & 3 \\ \hline 3 & 1 \\ \hline \end{array}$	C	$\begin{array}{ c c } \hline 1 & 2 \\ \hline 1 & r \\ \hline 2 & b \\ \hline 3 & j \\ \hline \end{array}$
-----	---	-----	---

.../8