

TD3 : XPATH

EX1: Lire

Objectif: Savoir interpréter des expressions XPATH

Q : Pour chaque élément de la liste suivante, évaluez l'expression XPATH en fonction du document XML associé.

- `/*/*/*BBB`

```

1<AAA>
2  <XXX>
3    <DDD>
4      <BBB/>
5      <BBB/>
6      <EEE/>
7      <FFF/>
8    </DDD>
9  </XXX>
10 <CCC>
11  <DDD>
12    <BBB/>
13    <BBB/>
14    <EEE/>
15    <FFF/>
16  </DDD>
17 </CCC>
18 <CCC>
19  <BBB>
20    <BBB>
21    <BBB/>
22    </BBB>
23  </BBB>
24 </CCC>
25</AAA>

```

Correction: 4, 5, 12, 13 et 20

- `/AAA/BBB[last()]`

```

1<AAA>
2  <BBB/>
3  <BBB/>
4  <BBB/>
5  <BBB/>
6</AAA>

```

Correction: 5

- `//BBB[not(@*)]`

```

1<AAA>
2  <BBB id = &quot;b1&quot;/>
3  <BBB id = &quot;b2&quot;/>
4  <BBB name = &quot;bbb&quot;/>
5  <BBB/>

```

6</AAA>

Correction: 5

- `//*[count(BBB)=2]`

1<AAA>

2 <CCC>

3 <BBB/>

4 <BBB/>

5 <BBB/>

6 </CCC>

7 <DDD>

8 <BBB/>

9 <BBB/>

10 </DDD>

11 <EEE>

12 <CCC/>

13 <DDD/>

14 </EEE>

15</AAA>

Correction: 10

- `//*[count(*)=3]`

1<AAA>

2 <CCC>

3 <BBB/>

4 <BBB/>

5 <BBB/>

6 </CCC>

7 <DDD>

8 <BBB/>

9 <BBB/>

10 </DDD>

11 <EEE>

12 <CCC/>

13 <DDD/>

14 </EEE>

15</AAA>

Correction: 1 et 2

- `//*[contains(name(),'C')]`

1<AAA>

2 <BCC>

3 <BBB/>

4 <BBB/>

5 <BBB/>

6 </BCC>

7 <DDB>

8 <BBB/>

9 <BBB/>

10 </DDB>

11 <BEC>

12 <CCC/>

13 <DBD/>

14 </BEC>

15</AAA>

Correction: 2, 11 et 12

- `//*[string-length(name()) = 3]`

```

1<AAA>
2  <Q/>
3  <SSSS/>
4  <BB/>
5  <CCC/>
6  <DDDDDDDD/>
7  <EEEE/>
8</AAA>

```

Correction: 1 et 4
- `/AAA/EEE | //DDD/CCC | /AAA | //BBB`

```

1<AAA>
2  <BBB/>
3  <CCC/>
4  <DDD>
5    <CCC/>
6  </DDD>
7  <EEE/>
8</AAA>

```

Correction: 1,2,5 et7
- `//DDD/parent::*`

```

1<AAA>
2  <BBB>
3    <DDD>
4      <CCC>
5        <DDD/>
6        <EEE/>
7      </CCC>
8    </DDD>
9  </BBB>
10 <CCC>
11  <DDD>
12    <EEE>
13      <DDD>
14        <FFF/>
15      </DDD>
16    </EEE>
17  </DDD>
18 </CCC>
</AAA>

```

Correction: 2, 4, 10 et 12

EX2: Ecrire

Objectif: Ecrire les expressions XPATH.

Q : Trouvez les expressions XPATH qui, évaluées au noeud racine de l'arbre donné, donne les node-sets suivants. Remarque: on donnera la version complète et la version abrégée ainsi que les numéros ou lettres des noeuds sélectionnés.

- toutes les sections

Correction:
`/descendant::section` ou `//section` soit 6, 11, 16, 21, 28, 32
- toutes les sections filles des éléments chapitres

Correction:

- `/descendant::chapitre/section //chapitre/section` 6, 11, 16, 30

toutes les sections contenant un paragraphe comme enfant direct

Correction:

`/descendant::paragraphe/parent::section` ou `//section[paragraphe]` soit 16
- toutes les éléments contenant un paragraphe comme enfant direct

Correction:

`/descendant::paragraphe/parent::*` ou `//paragraphe/..` soit 13, 16
- les parties de textes à gauche d'un acronyme

Correction:

`/descendant::acronyme/preceding-sibling::text()` ou `//acronyme/preceding-sibling::text()` soit 7, 18, 22, 25
- les titres des éléments contenant de manière directe un acronyme

Correction:

`/descendant::acronyme/parent::*/@titre` ou `//acronyme/../@titre` soit C, J, L
- les sections de niveau 2

Correction:

`/descendant::section[attribute::niveau=2]` ou `//section[@niveau=2]` soit les 21,28
- les titres des sections de niveau 2

Correction:

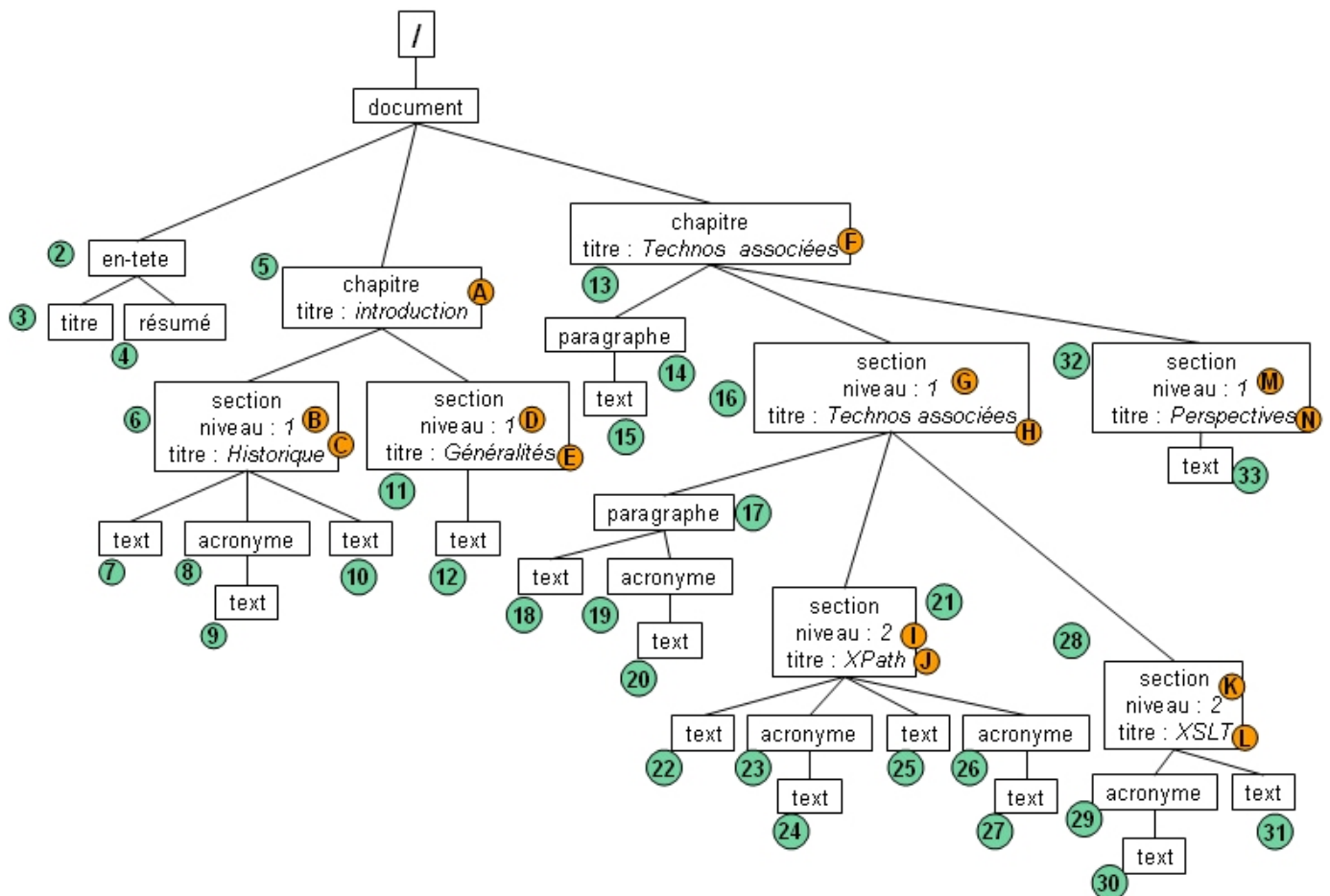
`/descendant::section[attribute::niveau=2]/attribute::titre` ou `//section[@niveau=2]/@titre` soit J, L
- les acronymes contenus dans la première section de niveau 2 du dernier chapitre

Correction:

`/descendant::chapitre[position() = last()]/descendant::section[attribute::niveau = 2][position() = 1]/child::acronyme` ou `//chapitre[last()]/section[@niveau = 2][position() = 1]/acronyme` soit 23, 26
- le titre des chapitres ne contenant aucun paragraphe

Correction:

`/descendant::chapitre[count(descendant::paragraphe) = 0]/attribute::titre` ou `//paragraphe[count(//paragraphe) = 0]/@titre` soit A



Q : Soit le document XML suivant

```
<CDlist>
  <CD>
    <composer>Johannes Brahms</composer>
    <performance>
      <composition>Piano Concerto No. 2</composition>
      <soloist>Emil Gilels</soloist>
      <orchestra>Berlin Philharmonic</orchestra>
      <conductor>Eugen Jochum</conductor>
    </performance>
    <performance>
      <composition>Fantasias Op. 116</composition>
      <soloist>Emil Gilels</soloist>
    </performance>
    <publisher>Deutsche Grammophon</publisher>
    <!--length is expressed in ISO standard duration format: 1h 13' 37" -->
    <length>PT1H13M37S</length>
  </CD>
</CDlist>
```

validant le fichier DTD suivant:

```
<!ELEMENT CDlist ((CD)+)>
<!ELEMENT CD (composer, (performance)+,
  publisher, (length)?)>
<!ELEMENT performance (composition, (soloist)?,
  (orchestra, conductor)?)>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
```

```
<!ELEMENT length (#PCDATA)>
<!ELEMENT composition (#PCDATA)>
<!ELEMENT soloist (#PCDATA)>
<!ELEMENT orchestra (#PCDATA)>
<!ELEMENT conductor (#PCDATA)>
```

Trouvez les expressions XPath qui retournent les informations suivantes (on suppose que le contexte initial est l'élément racine de nom CDlist):

- toutes les compositions
Correction: //composition (autre solution : /CDlist/CD/performance/composition)
 - toutes les compositions ayant un seul "soloist"
Correction: //performance[soloist]/composition Remarque:la DTD nous assure qu'il ne peut pas y avoir plus d'un soloist
 - toutes les performances avec un seul "orchestra" mais pas de "soloist"
Correction: //performance[not(soloist) and orchestra] Remarque:la DTD nous assure qu'il ne peut pas y avoir plus d'un orchestra
 - tout les soloists ayant joué avec le London Symphony Orchestra sur un CD publié par Deutsche Grammophon
Correction: /CDlist/CD[publisher="Deutsche Grammophon"]/performance[orchestra="London Symphony Orchestra"]/soloist
 - tout les CDs comportant des performances du London Symphony Orchestra
Correction: /CDlist/CD[performance[orchestra="London Symphony Orchestra"]]
- Correction: