

Transformations XSL (II)

Fabrice Rossi

12 mars 2003

Les exercices présentés dans ce document sont construits en partie grâce aux exercices de [1] et [2].

Exercice 1

On reprend dans cet exercice le dialecte inspiré de MathML proposé dans l'exercice 3.2 de [1]. Le format MathML ne correspond pas à l'écriture mathématique usuelle. On rappelle par exemple que $2 + 3$ s'écrit : `<apply><plus/><cn>2</cn><cn>3</cn></apply>`. Le but du présent exercice est d'écrire un programme XSLT qui transforme une expression MathML en une nouvelle expression dans l'ordre classique d'écriture mathématique. On veut par exemple que $2+3$ s'écrive : `<apply><cn>2</cn><plus/><cn>3</cn></apply>`. Pour simplifier l'écriture du programme XSLT, on ne change pas le nom des éléments, mais seulement l'ordre d'écriture. On souhaite par exemple que $(2 + 3) * 4$, qui s'écrit :

```
1 <apply>
2   <times/>
3   <apply>
4     <plus/>
5     <cn>2</cn>
6     <cn>3</cn>
7   </apply>
8   <cn>4</cn>
9 </apply>
```

soit transformé en :

```
1 <apply>
2   <apply>
3     <cn>2</cn>
4     <plus/>
5     <cn>3</cn>
6   </apply>
7   <times/>
8   <cn>4</cn>
9 </apply>
```

Pour réaliser le programme XSLT, on pourra utiliser l'opérateur *ou* (`|`). L'expression XPath `a|b`, par exemple, correspond à l'ensemble des éléments `a` et des éléments `b`, dans l'ordre du document. On pourra aussi utiliser la fonction `position()` qui permet de sélectionner des nœuds quand on la place dans un prédicat. Par exemple, l'expression XPath `a[position()=2]` correspond au deuxième élément `a` dans l'ordre de parcours du document.

Exercice 2

On reprend l'exercice 3.4 de [2]. Modifier le programme XSLT afin qu'il produise une page HTML donnant la liste des groupes et la liste des disques, avec des références croisées internes. Contrairement à ce qui est demandé dans l'exercice 3.4, le document HTML produit ne devra pas être redondant. Par exemple, chaque description de groupe ne contiendra qu'une liste de liens vers les descriptions de disques.

Pour gérer les références croisées, on utilisera l'élément **A** de HTML qui permet de définir une cible dont le nom est donné par l'attribut **NAME**. On utilisera aussi **A** avec l'attribut **HREF** pour définir un lien interne en prenant soit d'ajouter le caractère **#** devant le nom de la cible. Par exemple :

```
1 ...  
2 <A NAME="laCible">Voilà la cible</A>  
3 ...  
4 <A HREF="#laCible">Voilà le lien vers la cible</A>  
5 ..
```

Références

- [1] Fabrice Rossi. Dtd et schémas. Recueil d'exercices, Université Paris-IX Dauphine, 2002.
<http://apiacoa.org/teaching/xml/exercices/dtd-et-schemas.pdf>.
- [2] Fabrice Rossi. Transformations xsl. Recueil d'exercices, Université Paris-IX Dauphine, 2002.
<http://apiacoa.org/teaching/xml/exercices/xslt.pdf>.