Exercises 8

The purpose of these exercises is to get initial experience with languages for manipulating and querying XML data. The exercises are adapted from the web programming course at Lund University, taught by Thore Husfeldt. Data for the exercises can be found on the course schedule. Probably, part of the exercises will overflow to week 44.

Mindless XPath Drill

This exercise asks you to construct a number of XPath expressions to select specific parts of a given XML document. “linkedcollection.xml” is a version of the the large recipe collection of An Introduction to XML and Web Technologies by Møller and Schwarzbach, linked to the XSL stylesheets “selectstuff.xsl”.

Copy these two files into your own area. You are supposed to edit the second one. Open the XML file in a browser to see the results of running your XPath expressions.

Your task. Complete the remaining list items in the XSL stylesheet in the manner of the first two examples in the file. The exercise is supposed to force you through various features of XPath, so you might want to have that book handy.

XSL: Recipes by ingredient

This exercise asks you to construct an XSLT stylesheet for an alternative view of a recipe collection. Create a new copy of the recipe collection, and make it point to a stylesheet called “list-of-ingredients.xsl”. That is the only modification you may make to “linkedrecipes.xml”. Your task is to write “list-of-ingredients.xsl”.

Your task. The intended output of your XSL transformation presents the recipe collection by ingredient. With each ingredient, the list mentions the recipe(s) in which it appears. The list begins like this:

```xml
<h1>List of ingredients</h1>
<dl>
  <dt>Alchermes liquor</dt>
  <dd>Zuppa Inglese</dd>
  <dt>angel hair pasta</dt>
  <dd>Beef Parmesan with Garlic Angel Hair Pasta</dd>
  <dt>baking powder</dt>
  <dd>Ricotta Pie (dough)</dd>
</dl>
```

I have used an HTML “description list” DD for this, with ingredients in DT elements and recipe names in DD elements.
Warm-up
To begin, solve a simple sub-problem: make a list of all the top-level ingredients in the collection, i.e., those ingredients that are children of “recipe” elements. For example, the list should include “filling” but not “ricotta cheese”. Solve this problem in two ways (in two separate stylesheets):
1. Using xsl:apply-templates. This is called Declarative style (or, “Pushing, in XSL slang).
2. Using xsl:for-each. This is called Imperative style (or, “Pulling”, in XSL slang).

Further steps
In increasing order of complexity, your final solution needs to display the following features. It’s a good idea to solve these problems in this order.
1. The list must include all non-composite ingredients, e.g., “ricotta cheese” and even “mushroom juices”, but not “filling”
2. The list is sorted
3. The list includes with each ingredient the recipe in which it appears. So “Alechermes liquor” mentions “Zuppa Inglese”. (See the above example for a suggestion on formatting—but you can make it look however you want).
4. When an ingredient appears in a composite ingredient, this composite ingredient is named with the recipe. For example, the entry for “mushroom juice” looks like this: mushroom juices Cailles en Sarcophages (sauce)