Nesta aula de laboratório iremos resolver alguns exercícios relacionados com a detecção de duplicados em documentos XML, **considerando a estrutura dos documentos**.

Tenha em atenção que o software usado nas aulas de laboratório oferece, como complemento às funções da biblioteca XPath 2.0, uma **função de extensão para a medição de similaridade entre nós XML**, a qual utiliza uma *Rede Bayesiana* para combinar as similaridades entre cada um dos descendentes dos nós que se pretendem comparar. As similaridades entre os nós textuais são medidas através das funções apresentadas na aula de laboratório anterior (e.g. distancia de edição normalizada, a medida de Jaro, etc.). Para mais informações sobre o método utilizado, aconselha-se a consulta ao seguinte artigo:

- L. Leitão, P. Calado and M. Weis (2007) *Structure-based inference of xml similarity for fuzzy duplicate detection*. In Proceedings of the 16th ACM Conference on information and Knowledge Management, DOI= http://doi.acm.org/10.1145/1321440.1321483.

A função de extensão `gti.treesimilarity()` aceita três parâmetros de entrada. Os dois primeiros parâmetros correspondem aos nós XML que se pretendem comparar. O terceiro corresponde ao nó de raiz de um documento XML que apresenta a configuração para a *Rede Bayesiana*. A configuração define quais as funções de similaridade a utilizar na comparação dos nós textuais, assim como a probabilidade inicial associada a cada um dos nós descendentes (i.e., a similaridade para quando os nós não se encontram definidos).

A título de exemplo, considerem-se dois nós XML semelhantes ao nó da Figura 1.

```
<paper>
 <authors><author>Jianping Fan</author></authors>
 <locations><location>USA</location></locations>
 <title>Hierarchical Classification for Image Annotation</title>
 <abstract></abstract>
</paper>
```

Figura 1 – Um nó XML de exemplo contendo informação bibliográfica sobre artigos científicos.

Para comparar dois nós XML semelhantes aos que são apresentados na Figura 1 pode ser utilizada a configuração para a *Rede Bayesiana* que é apresentada na Figura 2. Esta configuração assume que todos os nós têm a mesma probabilidade inicial e que todos os nós textuais devem ser comparados usando a medida de Jaro.

```
<paper formula="Average">
 <authors defaultProbability="0.5" formula="Average" useFlag="true">
   <author defaultProbability="0.5" useFlag="true" similarityMetric="jaro" />
 </authors>
 <locations defaultProbability="0.5" formula="Average" useFlag="true">
   <location defaultProbability="0.5" useFlag="true" similarityMetric="jaro" />
 </locations>
 <title defaultProbability="0.5" useFlag="true" similarityMetric="jaro" />
 <abstract defaultProbability="0.5" useFlag="true" similarityMetric="jaro" />
</paper>
```

Figura 2 – Uma configuração de exemplo para a rede Bayesiana.

Para efectuar a comparação dos nós, e assumindo que existem variáveis com os nomes *$node1*, *$node2* e *$bconf*, pode ser utilizada a seguinte instrução XPath:

```
gti:treesimilarity($node1, $node2, $bconf)
```

Exercício 1

Identificar nós XML duplicados, sejam estes exactos ou aproximados, é um passo importante de um processo de **consolidação de dados semi-estruturados**. Por **duplicados aproximados** entendem-se os nós XML que não são exactamente idênticos mas que ainda assim são duplicados (e.g., dois nós podem diferir ligeiramente no conteúdo de alguns dos seus elementos constituintes).

A utilização de **funções de similaridade** é uma forma comum de abordar este problema. Dois registos com similaridade máxima correspondem a duplicados exactos, e dois registos com **elevada similaridade** têm uma **elevada probabilidade de serem duplicados**.

No URL https://dspace.ist.utl.pt/bitstream/2295/237039/1/cddb_data.xml encontra-se um documento XML contendo informação sobre discos compactos (CDs) e as músicas e artistas que se lhes encontram associados, contendo a descrição de 480 CDs escolhidos aleatoriamente da base de dados FreeDB (http://www.freedb.org/).

No URL https://dspace.ist.utl.pt/bitstream/2295/236816/1/cddb_dups.xml encontra-se um documento XML contendo informação sobre quais as descrições duplicadas de CDs que se encontram no primeiro ficheiro (i.e., os duplicados reais), sejam estes duplicados exactos ou aproximados.

1.1 – Escreva uma função em XQuery de nome `cd-similarity()` que permita encontrar quais os pares de CDs duplicados que se encontram no ficheiro `cddb_data.xml`, com base numa combinação linear de diferentes métricas de similaridade que permita considerar aspectos como o facto de os títulos de CDs poderem ser cadeias de caracteres semelhantes, ou os conjuntos de faixas poderem ter muitas sobreposições (i.e., através do *coeficiente de Jaccard* ou do *coeficiente de Dice*). Para cada par de CDs, a função XQuery deverá retornar a similaridade que lhe corresponde.

Neste exercício, deve ainda justificar a escolha das funções de similaridade com base nas propriedades que conhece sobre estas funções.

1.2 – Defina um ficheiro de configuração para a rede Bayesiana a ser utilizada na função `gti.treesimilarity()` por forma a medir a similaridade entre os discos compactos definidos no ficheiro `cddb_data.xml`. Justifique a escolha das funções de similaridade usadas para comparar os nós textuais, com base nas propriedades que conhece sobre as diferentes funções de similaridade.

1.3 - Escreva uma XQuery que, utilizando o resultado da alínea anterior, permita calcular a similaridade entre as diferentes definições de discos compactos que se encontram no ficheiro `cddb_data.xml`. Para cada par diferente de CDs, a XQuery deverá retornar a similaridade que lhe corresponde.

1.4 - Escreva uma XQuery que, usando a função `gti.treesimilarity()`, para calcular as similaridades em conjunto com o mesmo mecanismo da função `local:distinct-nodes()`, apresentada numa aula de laboratório anterior, remova do ficheiro `cddb_data.xml` as definições de CDs correspondentes a duplicados aproximados (i.e., para os CDs duplicados, o XML produzido como resultado deverá conter apenas 1 CD). Por duplicados aproximados, entende-se que devem ser considerados os pares de CDs com uma similaridade combinada superior a 0.9.

1.5 – Usando o ficheiro XML contendo informação sobre quais os elementos que são duplicados reais, infira a qualidade dos resultados obtidos na alínea 1.4. Para tal, deverá escrever uma XQuery que contabilize quantos dos duplicados detectados são duplicados reais (i.e., medir a **precisão**), e quantos dos duplicados reais são detectados (i.e., medir a **abrangência**). Considere mais uma vez que os duplicados aproximados correspondem aos nós com uma similaridade combinada superior a 0.9.

1.6 - Escreva uma XQuery que permita consolidar a informação no documento XML com descrevendo os CDs, fundindo os CDs que sejam duplicados num único registo. Ao fundir os CDs duplicados, deve procurar obedecer aos seguintes critérios:

- Preencher os vários elementos, combinando a informação dos CDs duplicados.
- Manter o título mais completo (i.e., o título correspondente á maior *string*)