Dokumentace úlohy XQR: XML Query v Python 3.6 do IPP 2016/2017
Jméno a příjmení: Petr Jůda
Login: xjudap00

1 Úvod

1.1 Rozbor zadání

Cílem projektu bylo vytvořit script v jazyce Python, který na svém vstupu načte XML soubor, pomocí zadaného query dotazu provede jeho vyfiltrování a na výstupu vygeneruje XML soubor, obsahující pouze elementy vyhovující zadané podmínce. Script pracuje v několika krocích popsaných níže.

2 Implementace

2.1 Zpracování parametrů

Celý proces začíná vyhodnocením a zpracováním zadaných vstupních parametrů. Po spuštění scriptu je volána funkce `parseargs()`, která nastaví přípustné přepínače pro knihovnu `argparse`. Následně je kontorolováno, zda parametr `--help` není kombinován s ostatními přepínači nebo se některé neopakují. Pokud je vše v pořádku a operace `parse_args()` skončila úspěchem jsou zadané hodnoty uloženy do slovníku. Je-li zadán přepínač `--qf`, pak je dotaz načten ze souboru.

2.2 Zpracování dotazu

Po zpracování vstupních parametrů je volána funkce `checkQuery()`, která validuje a zpracovává jednotlivé položky dotazu. Nejprve je všem hledaných položkám přiřazena implicitní hodnota `None`. Následně je dotaz zpracováván jako běžný textový řetězec zleva doprava. Vždy jsou pomocí funkce `.lstrip()` odstraněny bílé znaky z levé strany řetězce. Následně je pomocí standardní řetězcové funkce `.find()` zkontrolováno, zda a případně na jakém místě se vyskytuje požadované nebo volitelné klíčové slovo. Po nalezení pozice je hodnota zkopírována do slovníku a dotaz následně oříznut. Tato operace se opakuje, dokud není zpracován celý dotaz. Speciálně je zpracováváno klíčové slovo `NOT`, které se může vyskytovat opakovaně. Při každém výskytu je zvýšeno číselné počítadlo, jehož výsledný zbytek po dělení dvěma určí, zda se má zavolat funkce `checkNot`, která neguje relační operátor v dotazu.

2.3 Získání vstupního souboru

Načtení vstupního souboru je možné ze standardního vstupu nebo z uloženého souboru. Při první zmíněné variantě je nejprve celý soubor uložen do textové proměnné. Následně je pomocí metody `parseString()` z knihovny `xml.dom.minidom` vytvořen objekt reprezentující vstupní XML. V druhém případě je proveden pokus o otevření vstupního souboru a data jsou načteny pomocí metody `parse()` z již zmiňované knihovny.

2.4 Filtrování vstupního souboru

Po získání všech potřebných údajů je volána funkce `filterXML()`, která řídí proces fitrování. Nejprve je proveden pokus o nalezení kořenového elementu z klauzule `FROM`. Pokud se v klauzuli nachází element nebo element.atribut je pomocí knihovních funkcí `getElementsByTagName()` a `hasAttribute()` nalezen kořenový element. Pokud je zadán pouze atribut je volána funkce `invokeRecursion()`, která provádí `recursionSeach()` jehož výsledkem je list obsahující všechny elementy dokumentu, kde je následně pomocí `hasAttribute()` nalezen kořenový element. Pokud takový element neexistuje, je vytvořen prázdný soubor. Pokud dotaz obsahoval klauzuli `WHERE` je volána funkce `processWhere()`, která pro každý podelement nebo atribut pomocí funkce `testCondition()` kontroluje, zda vyhovuje zadané podmínce. Pokud ano je element přidán do výsledného listu `final`, jehož obsah se objeví ve výsledném XML. V případě přítomnosti klíčového slova `LIMIT`, je velikost listu redukována na počet zadaný v dotazu.

2.5 Generování výstupu

V závislosti na vstupních parametrech je výsledné XML generováno na standardní výstup nebo do souboru. Obsah listu `final` je vypsán pomocí funkce `toxml()`. Pokud byly zadány přepínače `-n` nebo `--root` je výsledné XML doplněno o kořenový element, případně není generována XML hlavička.