

Popis problému

Úlohou bylo vytvořit skript v jazyce Python, který vyhodnocuje zadaný dotaz, podobný dotazu SELECT jazyka SQL, nad vstupním XML souborem. Skript postupně ověřuje vstupní argumenty, dále kontroluje validitu dotazu a nakonec vyhodnotí dotaz nad XML souborem.

Zpracování argumentů

Na zpracování argumentů jsem si vytvořil třídu *Options*. V této třídě používám třídu *AParser*, která dědí z třídy *argparse.ArgumentParser* a přepsal jsem metodu `error()`. Vytvoření této třídy a přepsání metody `error()` bylo potřebné na umlčení výpisů třídy *argparse.ArgumentParser* na výstup. Tím jsem si vytvořil možnost vlastního zpracování chyb argumentů.

Hodnoty z argumentů jsou zpracované a ukládané do kolekce dictionary. Přepínače input a output nesou v této kolekci otevřené popisovače souborů, implicitně to jsou stdin/stdout. Proto po ukončení skriptu jsou tyto popisovače zavřené. Pokud byl zadaný přepínač `qf`, získá se obsah tohoto souboru a dále se s ním bude pracovat jako s přepínačem query.

Validace dotazu

Pro validitu dotazu mám opět vytvořenou třídu *Query*. Jelikož v zadání máme kompletní gramatiku pro dotaz, tak jsem do této třídy implementoval rekurzivní syntaktickou analýzu. Abych jí mohl bez problémů použít, tak jsem si dovolil trochu upravit dotaz, přidáním mezer na místa, kde nejsou povinné mezery okolo znaků jako jsou `<`, `>`, `=` a textový řetězec. Následně po přidání těchto mezer jsou všechny bílé znaky nahrazené právě jednou mezerou. Tím mám vytvořený oddělovač tokenů pro zmíněnou analýzu. Během analýzy kontroluji čísla, identifikátory a textové řetězce na správný formát. Potřebné informace jsou z dotazu extrahované do kolekce dictionary.

Selekce

Na vyhodnocení dotazu je určená třída *Selection*. V ní používám třídu *xml.dom.minidom* na zpracování XML dokumentu, se kterým se bude pracovat.

V první fázi se určuje počáteční element, kořenový, z klauzule FROM. Dále následuje výběr všech elementů od tohoto kořenového elementu z klauzule SELECT. Tyto dvě fáze se vykonají vždy, pokud je validní dotaz a klauzule FROM není prázdná. Další klauzule WHERE a LIMIT jsou volitelné. Klauzule WHERE se vyhodnocuje ve 2 částech. První část, podmínka, rozdělí elementy na listy s elementy podle výsledku na true a false. Druhá část rozhodne, který z těchto listů splní podmínku doplněnou o případnou negaci NOT. Nakonec klauzule LIMIT je vyhodnocená v metodě `get_elements()`, kde, pokud byla zadaná klauzule LIMIT, se ořeže list na prvních N elementů, jinak se vrátí celý list.

Formátování výstupu

Výstup je podle přepínačů doplněný o potřebné informace, jako je hlavička XML dokumentu, či kořenový element. Pokud pomocí dotazu byl vybrán alespoň jeden element, tak je nazpět převedený do textové podoby pomocí metody `toxml()` třídy *xml.dom.minidom*. Finálním ukončením výstupu je doplnění výstupu o znak konce řádku.