

Úvod:

Můj způsob řešení úlohy XQR: XML Query se skládá z několika částí. Jádrem celého programu je funkce `execute()`, která spouští další funkce ve správném pořadí, a do jisté míry zajišťuje i přehlednost zdrojového kódu. Důležité hodnoty pro správnou činnost programu se ukládají do 3 polí, `$params` a pole `$flags` a `$queryElems`. Do pole `$params` se ukládají informace načtené z argumentů použitých při spuštění programu, s tímto polem souvisí i pole `$flags`, které uchovává `Boolean` hodnoty pro použití jako přepínače. Poslední pole `$queryElems` reprezentuje dotaz, který je ale rozparsovaný na části a obsahuje klíče pro lehký přístup ke všem hodnotám.

Zpracování vstupních argumentů:

První je potřeba zpracovat vstupní argumenty. Funkce `handleArgs()` prochází globální pole `$argv` a každou položku porovnává s možnými argumenty programu. V případě shody se volá funkce `checkFlags()`, která kontroluje požadavky ze zadání, vícenásobné zadání argumentů a povolené kombinace argumentů. V zadání XQR nebylo specifikováno jaké zkrácené argumenty má program podporovat. Proto jsem se rozhodl implementovat zkrácené argumenty pro všechny možnosti kromě `-query` a `-qf`. Způsob zadávání je pomlčka a první písmeno argumentu. Pokud se v poli `$argv` vyskytne řetězec který se s žádnou možností neshoduje, funkce zahlásí chybu a program se ukončí. Pokud je zadán jediný argument, `-help`, spustí se funkce `help()`, která vypíše krátkou nápovědu k možným argumentům a program se ukončí.

Ověření správnosti vstupních souborů a query:

V jiném případě se zavolá funkce `checkfiles()`, podle argumentů zkontroluje správnost zadaného vstupního i výstupního souboru. Následuje funkce `getQuery()`, ta uloží dotaz do proměnné `$query` pro pozdější zpracování, v případě argumentu `-qf` musí načíst data ze vstupního query souboru. Tato `$query` se hned v následujícím okamžiku využije ve funkci `validateQuery()`. Ta nejen ověří správnost dotazu podle bezkontextové gramatiky vypsáné v zadání úlohy, ale také naplní pole `$queryElems` jednotlivými lexémy. V implementaci funkce `validateQuery()` jsem použil porovnání klíčových slov pomocí regulárních výrazů. V případě shody se následující lexém načítá pomocí funkcí `getElOrAt()`, `getLiteral()`, `getRelOp()` nebo `getCondition()`. Pro testovací účely se volá i funkce `queryAppend()`, která se hodí v případě nalezení chyby v načítání dotazu.

Vykonání dotazu nad XML souborem:

Po úspěšné kontrole argumentů, načtení vstupních souborů a kontrole dotazu se může začít se samotným zpracováním vstupního XML souboru. Pro potřeby mé implementace jsem vytvořil 2 zpracovávací funkce (pre-funkci pro předzpracování vstupů a vytvoření prostoru pro výstup, a funkci která je z této pre-funkce volána a provádí výběr elementů) pro téměř každé klíčové slovo dotazu. Tedy pro `SELECT`, `FROM`, `WHERE`, `ORDER` a `LIMIT`. Pouze pro `FROM` a `SELECT`, které vykonávají stejnou věc, používám 1 funkci, ale s jinými argumenty. Abych získal hlavní skupinu elementů, která je určená za klíčovým slovem `FROM`, zavolám tedy funkci `select($queryElems["from"], $queryElems["root"], $xml, "from")`, kde specifikuji hledaný element, zdrojový element, vstupní XML a slovo "from", které je pouhý flag, aby se výstupní `SimpleXMLElement` skládal pouze z prvního výskytu hledaného elementu. V této pre-funkci vytvořím XML hlavičku, zavolám zpracovávací funkci `$findEl($queryEl, $src)`, která rekurzivně prohledává `$queryEl` dokud se momentálně porovnávaný element neshoduje ve jméně, atributu, nebo jméně i atributu podle zadaného dotazu. V případě shody uloží tento prvek do pole `$xmlDom`. Toto pole si přebere pre-funkce, spojí ho s XML hlavičkou, Načte do nového `SimpleXMLElement` a vrátí ho. Nad tímto `SimpleXMLElement` se postupně provádí všechny další operce, tedy `SELECT`, a v případě potřeby i `WHERE`, `ORDER-BY` a nakonec i `LIMIT`. S implementací `WHERE NOT` jsem měl nemalé potíže, a proto jsem se ji rozhodl udělat stylem doplňku, kdy nejdříve provedu obyčejný `WHERE` a poté odstráním z hlavního elementu všechny elementy, které se našly pomocí `WHERE`. Výsledný `SimpleXMLElement` se ještě před zapsáním do output souboru ošetří pomocí regexového nahrazení.