Task documentation: XQR: XML Query in Python 3 for IPP 2011/2012
Name and surname: Michal Lúkáč
Login: xlukac05

Introduction

This program does the evaluation of query similar to SELECT statement from SQL language over xml files or stdin with xml format. The implementation is in Python 3 programming language which is strong for natural languague processing, basic scripting, hacking and web developing. Python 3 supports also object-oriented and functional programming styles.

Arguments processing

If you do not know how to use this script, run program with --help switch. For script arguments, program doesn't use module. Script arguments are processed in function procParams. I used global variable G_MYDICT which represents dictionary type from Python 3. In this variable program stores information about generating xml-header, root element name or inputfile/stdin.

Implementation

After what arguments are processed and flags are set, the main function programm calls prepareText. This function removes all commentaries from xml text and also substitute some chars for lexical analysis. After this, function searchTokens() is called. In this function simple finite-state machine handles text of select query and get tokens from it. When program get tokens, simple syntax and semantics analysis is done in syntaxAnal() function. For syntax analysis I use grammar rules, which were defined in project definition. This rules tell us things like first token must be "SELECT". "SELECT" is followed by element and element is followed by "FROM"... If the structure of query is bad, program prints error message and exit with value 80. Program save SELECT and FROM element to dictionary variable G_MYDICT. Next function parseXML() use parseString() from module minidom for xml parsing. With this module I can effectively search elements and attributes with functions getElementsByTagName(), childNodes(), hasAttribute(). Function goThroughXml() is for deep searching of elements. When WHERE token occurs parseXML() calls whereSolve() which do precedence analysis and evaluate WHERE clause over each element and return True or False. If True element is printed to stdout.

Precedence analysis

For this script I do extension LOG with which script can fold conditions with AND, OR operators. For example: 'SELECT item FROM ROOT WHERE NOT NOT NOT name CONTAINS "Lorem Ipsum" AND id > 30' is possible query. I do this with precedence analysis which is suitable for expression evaluation. Precedence analysis is implemented with stack and precedence table. Precedence table is represented in script as 2-dimensional array and stack is represented by in-built python3 class list with functions pop() and append(). ItemTable is class which consist of token and number of position in table. Function mapToTable(token), return index of token to table. Function whereSolve(node) push to or do reduction from stack accordnig to last item on stack and input token. If program want to know next action(shift/reduction), first program must look to

	AND	OR	NOT	(	)	OP	EL	$
AND	>	>	<	<	>	<	<	>
OR	<	>	<	<	>	<	<	>
NOT	>	>	<	<	>	<	<	>
(	<	<	<	<	>=	<	<	X
)	>	>	>	X	>	X	X	>
OP	>	>	X	<	>	X	<	>
EL	>	>	>	X	>	<	X	>
$	<	<	<	<	X	<	<	X

precedence table and get action(< is shift to stack, > reduce from stack) from table. If program gets from table ‘X’, program ends with error. Function isKeyword(token) returns True if token in WHERE-CLAUSE is keyword. This leads again to syntax/semantics error with return value 80.