DOMAIN-SPECIFIC LANGUAGE FOR CITATION GENERATION

Andrei BERCO, Simion CUZMIN*, Olivia GINCU,
Daniel MUNTEAN, Victor REVENCO

Department of Software Engineering and Automatics, Group FAF-221, Faculty of Computers, Informatics and Microelectronics, Technical University of Moldova, Chișinău, Republic of Moldova

*Corresponding author: Simion CUZMIN, simion.cuzmin@isa.utm.md

Scientific coordinator: **Irina COJUHARI**, conf. univ., dr., Technical University of Moldova

Abstract. *In this paper, a new Domain Specific Language called CiteGen is proposed, which will make it easier to create citations in a variety of forms. The document explains the grammatical and syntactical nuances of CiteGen as well as the specifics of its implementation, giving readers a thorough understanding of the framework's functionality. In addition, the study suggests future areas for CiteGen development and advances possible paths for its improvement. CiteGen is marketed as an easy-to-use tool that provides writers with a productive way to automate citation operations.*

Keywords: analysis, citations, grammar, implementation, academic research

Introduction

Scholarly conversation is a fundamental component of information sharing, helping to spread discoveries, advances, and new ideas throughout different fields. The careful and moral inclusion of citations, which not only recognize the contributions of others but also provide writers' claims context and validity, is essential to this process. Making citations is a difficult undertaking, though, as there are many distinct citation styles to choose from and formatting requirements set forth by various academic organizations and publication standards. This Domain-Specific Language seeks to improve accuracy, expedite citation processes, and maintain compliance with established citation standards, particularly those delineated by reputable organizations. Additionally, this paper clarifies the DSL's grammar, lexical quirks, semantic rules, data structures, and control methods so that users can effectively utilize it. It also offers techniques for structuring mathematical formulas in the DSL such that mathematical statements are consistent and understandable across academic writings.

Domain Analysis

A variety of genres are included in scientific writing, such as reviews, technical reports, and research papers. It is distinguished by objectivity, objectivity, and clarity, with an emphasis on providing research findings and bolstering claims with proof from reliable sources. Quotations can be used to bolster claims, offer proof, or clarify ideas. Quoting should be done sparingly, though, and whenever feasible, information should be summarized or paraphrased rather than directly quoted. Quoted content needs to be correctly acknowledged to the original author, adhering to the guidelines specified by the selected citation style. Maintaining coherence and flow in scientific writing requires the smooth integration of quotes. Contextualization, signal words, and transitions aid in introducing and tying quotes to the body of the text. It is best to use quotes seamlessly, avoiding sudden or jerky changes from the writer's words to the cited content. The goal of a Language for auto-generating quotes for scientific articles is to make the process of adding quotes to research manuscripts more efficient. Functionalities for choosing pertinent quotes from source materials, formatting quotes in accordance with the selected citation style, and incorporating quotes into the text with the proper credit and context would all need to be included in this DSL. Researchers can maintain uniformity in their quotation methods and save time by

implementing automation. Although there aren't always international standards that are only for quoting in scientific works, a number of academic and publishing organizations offer citation and reference criteria that cover the use of quotes.

Table 1 presents five popular standards for citation and their description.

Table 1

General overview of citation standards

American Psychological Association	When quoting passages shorter than 40 words, enclose them in quotation marks and seamlessly integrate them into your own writing. When formatting quotations of 40 words start the block quotation on a new line and indent the entire block by 0.5 inches from the left margin [1].
Modern Language Association	When quoting a source in paper, it will include the author's last name and the page number of the quote. In the bibliography, the entry for the source typically begins with the author's last name, which corresponds to the name used in the parenthetical reference [2].
Chicago Manual of Style	For in-text citations in Chicago style, Notes and Bibliography formatting necessitates the use of footnotes and endnotes to acknowledge various sources utilized in the work. When referencing a source within the text, a roman numeral is inserted at the end of the borrowed information as a superscript. This number corresponds to a footnote or endnote [3].
Council of Science Editors	In-text citations in CSE is followed a simple numbering system, where a superscripted number at the end of a clause or sentence denotes external source material. These numbers correspond to entries in the References list at the end of the document [4].
ISO 690:2012	ISO 690:2012 offers comprehensive guidelines for referencing various sources, including electronic documents. It provides a structured framework for organizing citation elements like author names, publication titles, dates, and page numbers. ISO 690:2012 offers guidelines for author name presentation, formatting publication titles, necessary publication details, formatting URLs or DOI links for electronic resources, punctuation, and capitalization within citations [5].

The development of a software language that generates the source of quotes in scientific texts based on user input involves several key stakeholders. Each stakeholder plays a crucial role in ensuring the success, usability, and ethical use of the software. Some stakeholders, as well as the target groups of this Domain Specific Languages are developers, educational institutions, publishers, standard organizations and quality assurance teams. The time-consuming aspect of manually formatting citations is one of the main issues that needs to be resolved with automated citation production. Writing precise citations in the required format can be a tedious effort for academics, students, and writers, and it frequently requires careful attention to detail. This procedure entails recognizing different sources, including books, journal articles, websites, and more, and then following the formatting guidelines specified by the selected citation style. By automating this procedure, writers can greatly cut down on the time and effort needed, spending more of their time on writing and research and less time on formatting details. The primary goal of the DSL is to enhance the efficiency of the citation process for researchers, academics, and writers. By automating the generation of citations, users will be able to save time and effort, allowing them to focus more on their research and writing tasks. The Domain Specific Language will ensure the accuracy and consistency of citations by adhering to established citation standards.

By following predefined rules and formatting guidelines, the Domain Specific Language will minimize errors and inconsistencies in citation formatting, thereby improving the quality and credibility of scholarly documents. The Domain Specific Language will provide flexibility in terms of citation styles and formatting options. Users will have the ability to choose from a range of citation styles commonly used in academic writing, and customize citation elements such as author names, publication dates, and page numbers to suit their specific requirements.

Grammar

An example of a grammar that could help create the Domain Specific Language [6] for citation generation is given below.

Table 2 presents meta-notations used in grammar definition.

Table 2

Meta – notations

<foo>	Means foo is a non-terminal symbol
foo	(in bold font) means foo is a terminal i.e., a token or a part of a token
[x]	Means zero or one occurrence of x i.e., x is optional. Note that '[' and ']' are terminal
x*	Means zero or more occurrences of x
x^+,	A comma-separated list of one or more x's
{}	Large braces are used for grouping; note that braces in quotes '{' '} ' are terminals.
|	Separate alternatives

<program> ::= main <block>
<block> ::= <var_decl>* <statement>*
<var_decl> ::= <type> <id>$^+$,
<type> ::= **int | string | bool | dict**
<statement> ::= <expr> | **if** <expr> : <block> [**else** : <blok>] | **for** <id> **in** <expr> : <block> | **break** | <block> | <method_call>
<method_call> ::= <method_name>([<expr>*])
<method_name> ::= <id> | **CiteAPA | CiteMLA | CiteCMS | CiteCSE | CiteIEE | CiteISO**
<expr> ::= <method_call> | <literal> | [<type>] <expr> <operation> <expr> | **print**(<expr>) | <id>
<operation> ::= <arithm_op> | <rel_op> | <eq_op> | <cond_op> | <assignment_op>
<arithm_op> ::= + | - | * | /
<rel_op> ::= < | <= | > | >=
<eq_op> ::= == | !=
<cond_op> ::= **and | or**
<assignment_op> ::= = | +=
<literal> ::= <int_literal> | <char_literal> | <bool_litearal> | <string_literal>
<id> ::= <char> {<char_literal> | <int_literal>}*
<char> ::= a-zA-Z
<digit> ::= 0-9
<int_literal> ::= <digit><digit*>
<bool_literal> ::= **True | False**
<char_literal> ::= ' <char> '
<string_literal> ::= " <char>* "
<dict> ::= <literal> : <literal>

Some important aspects of the grammar are:

1. All keywords are lowercase. Keywords and identifiers are case-sensitive. For example, if is a keyword, but IF is a variable name; foo and Foo are two different names referring to two distinct variables.
2. The reserved keywords are: **and, bool, break, CITE, char, def, dict, else, False, for, if, in, int, or, return, string, True, print, CiteAPA, CiteMLA, CiteCMS, CiteCSE**
3. Comments are started with # and are terminated by the end of the line
4. White space may appear between any lexical tokens. White space is defined as one or more spaces, tabs, page and line-breaking characters, and comments.
5. Keywords and identifiers must be separated by white space, or a token that is neither a keyword nor an identifier.
6. String literals are composed of ’s enclosed in double quotes. A character literal consists of an enclosed in single quotes. Numbers are 32 bits signed. That is, decimal values between -2147483648 and 2147483647. Dictionary represents a list of pairs (key, value).
7. A is any printable ASCII character (ASCII values between decimal value 32 and 126, or octal 40 and 176) other than quote (”), single quote (’), or backslash (\), plus the 2-character sequences” \” to denote quote,” \’” to denote single quote,” \\” to denote backslash,” \t” to denote a literal tab, or” \n” to denote newline.

Parsing tree

In computer science and linguistics, a parsing tree—also called a syntax tree or a derivation tree - is a basic idea. It functions as a graphic representation of a string of symbols' syntactic structure as determined by a formal language [7].

Bellow are shown a code snippet according to the grammar described previously and its parsing tree represented in Fig.1.

```
start
string[3] quotes
quotes = ["First Quote", "Second Quote", "Third Quote"]
for quote in quotes:
CiteMLA(quote, "book", "John Doe")
```

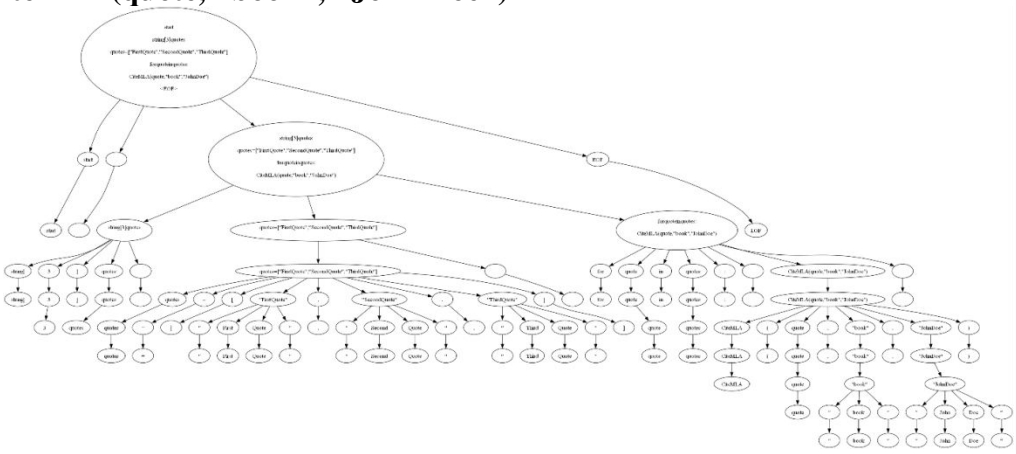

Figure 1. Example of parsing tree

Conclusion

To sum up, this thorough investigation explores the domain analysis of a language designed specifically to automate the creation of citations in scientific publications. It covers essential elements of scientific writing, citation standards from reputable organizations like APA, MLA, CMS, and CSE, and emphasizes the need of moral quoting procedures. The project intends to improve efficiency, correctness, and integration in academic and scientific writing processes by

optimizing citation creation, hence meeting the requirements of a wide range of users in many disciplines. Additionally, the document offers thorough explanations of the Domain-Specific Language's syntax, lexical considerations, semantic rules, data kinds, and control statements. These rules guarantee lucidity, accuracy, and consistency, empowering users to make the most of the DSL and improve efficiency and honesty in academic writing.

References

[1] "Quotations" [Online]. Available: https://apastyle.apa.org/style-grammar-guidelines/citations/quotations
[2] "MLA (Modern Language Association) Style: In-text citations" [Online]. Available: https://guides.library.uab.edu/MLAStyle/intext#:~:text=For%20prose%20quotes%20of%20f
[3] "The Ultimate Guide to Citing Anything in Chicago Style" [Online]. Available: https://www.citationmachine.net/chicago
[4] "Citation Guide: Council of Science Editors (Citation-Sequence System)" [Online]. Available: https://wac.colostate.edu/repository/writing/guides/cse/
[5] "Reguli pentru prezentarea referințelor bibliografice" [Online] Available: Referinte_bibliogr.pdf (utm.md)
[6] G. Karsai, H. Krahn, C. Pinkernell, B. Rumpe, M. Schindler and S. Völkel, "Design Guidelines for Domain Specific Languages", In Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (DSM' 09) Helsinki School of Economics. TR no B-108. Orlando, Florida, USA, October 2009 [Online] Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/ftp/arxiv/papers/1409/1409.2378.pdf
[7] "Parse Tree in Compiler Design" [Online]. Available: https://www.geeksforgeeks.org/parse-tree-in-compiler-design/