Linguistics Foundations of Computer Science and Its Applications

Azhari Qismallah

Computer Science Department, University College at Umluj, University of Tabuk

Abstract
The paper aimed at recognizing an understanding the state-of-the-art linguistics foundations of computer science and its applications that affect numerous computer science knowledge areas and their body of knowledge. The linguistics foundation, along with others, allows designing curricula that make computer science audiences acquiring the capability to recognize and understand the linguistics ideas that they can employ in their contributions on the field and graduate them with the required characteristics and experiences. The paper used literature analysis method to conduct its descriptive research. The paper findings show that the linguistics is a foundational material for at least half of computer science knowledge areas.

Keywords: *Linguistics, Computer Science, Body of Knowledge, Knowledge Area.*

.I. INTRODUCTION

Computer science is an interaction among theorem, coding, machine, and applications [1]. Linguistics is the one of foundational materials for computing. "Computing is a broad field that connects to and draws from many disciplines, including mathematics, electrical engineering, psychology, statistics, fine arts, *linguistics*, and physical and life sciences [1]."

“Discrete structures are foundations of computer science. With foundational material we signify that moderately barely any computer researchers will be functioning principally on discrete structures, yet that numerous different computer fields require the capacity to work with ideas from discrete structures. Discrete structures contain significant material from such zones as set theorem, logic, graph theorem, and probability theorem. [1]" According to the two previous quotes, the following analogy can be made among discrete structures, *linguistics*, and all other stated foundational material for computing. “*Linguistics is foundational material of computer science. With foundational material we signify that moderately barely any computer scientists will be working principally on linguistics, yet that numerous different computer fields require the capacity to work with ideas from linguistics*.”

The main questions of the paper main are to *what extent can the knowledge of linguistic foundations impact computer science field and its applications?* In the other words*, what are the critical roles that linguistic foundations can play in computing area and its applications?* Or, *to what extent the students and other audiences need to understand linguistics foundations for computing field and its application?*

The paper examines Computer Science Curricula 2013 [1] in order to interrogate computer science knowledge areas and their body of knowledge, and identifying the important factors from the computer science curricula (inclusion of linguistics material within computer science knowledge area or inclusion of linguistics material within body of knowledge).

The paper presents descriptive topic, and it uses literature analysis method to acquire a comprehension of the state-of-the-art linguistics foundational materials for computing and its applications.

Section II introduces the subject background, whereas Section III introduces linguistics material that are included within computer science body of knowledge. Section IV introduces linguistics foundation for specific knowledge areas in computer science. Finally, the paper concludes with Section V, which presents the conclusion.

II. BACKGROUND

The section presents fundamental knowledge necessary in order to understand the problem of linguistics foundational materials for computing and its applications.

Knowledge Area (KA). "A Knowledge Area represents a complete set of concepts, terms, and activities that make up a professional field, project management field, or area of specialization." [2]. Like other curriculum, the Body of Knowledge- the title of themes that ought to show up in undergrad computer science curriculum- in computer science curricula is coordinated on a group of knowledge areas. In ACM/IEEE computer science curricula 2013, “The CS2013 Body of Knowledge is coordinated into a group of (18) Knowledge Areas (KAs), relating to subject fields of education in computing, are coordinated on subjects as opposed to by course limits. Every KA is additionally coordinated into a group of Knwoledge Units (KUs), which are summed up in a table at the top of every KA area.” [1].

Manuscript received October 5, 2021
Manuscript revised October 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.10.39

Boy of Knwoledge (BoK). The Computer Science Body of Knowledge describes the knowledge within computer science disciplines. "In Computer Science terms, one can see the Body of Knowledge as a detail of the subject to be taught and syllabus as an implementation." [1].

This paper investigates each KUs (that are stated in ACM/IEEE Computer Science Curricula 2013), determines whether the linguistics is a foundational material for them or not; that is, in order to recognize and understand the impact of linguistics foundations in computer science and its applications.

III. INCLUSION OF LINGUISTICS MATERIAL WITHIN THE BODY OF KNOWLEDGE OF COMPUTER SCIENCE CURRICULA

As mentioned in previous section, The CS2013 Body of Knowledge is organized in a group of (18) Knowledge Areas (KAs), relating to subject fields of study in computing, coordinated on subjects as opposed to by course limits. Every KA is additionally coordinated into a group of Knwoledge Units (KUs). Referring to [1], the section reviews each KUs within eighteen Knowledge areas, which in turn construct the Body of Knowledge, that the foundational material of them is linguistics.

Table I through Table IX below view the various KU's topics that the linguistics is a foundational material for them.

TABLE I. LINGUISTICS FOUNDATION PER KUs' TOPICS IN ALOGRITHM AND COMPLIXTY KA

Knowledge Unit	*Topic*
Basic automata, computability and Complexity	Context-free grammars (cross-reference PL/Syntax Analysis)
Advanced automata theory and computability	Sets and languages, Context-free languages
	Chomsky hierarchy

TABLE II. LINGUISTICS FOUNDATION PER KUs' TOPICS IN ARCHITECTURE AND ORGANIZATION KA

Knowledge Unit	*Topic*
Machine Level Representation of Data	Representation of non-numeric data (character codes, graphical data)
Assembly Level Machine Organization	Instruction sets and types (data manipulation, control, I/O)
	Assembly/machine language programming

TABLE III. LINGUISTICS FOUNDATION PER KUs' TOPICS IN COMPUTATIONAL SCIENCE KA

Knowledge Unit	*Topic*
Introduction to Modeling and Simulation	Simulations as dynamic modeling
	Simulation techniques and tools, such as physical simulations, human-in-the-loop guided simulations, and virtual reality
Modeling and Simulation	The simulation process; identification of key characteristics or behaviors, simplifying assumptions; validation of outcomes

TABLE IV. LINGUISTICS FOUNDATION PER KUs' TOPICS IN HUMAN-COMPUTER INTERACTION KA

Knowledge Unit	*Topic*
Foundations	Contexts for HCI (anything with a user interface, e.g., webpage, business applications, mobile applications, and games)
Designing Interaction	Help and documentation
Programming Interactive Systems	Choosing interaction styles and interaction techniques
User-Centered Design and Testing	Functionality and usability requirements (cross-reference to SE/Requirements Engineering)
	Techniques for gathering requirements, e.g., interviews, surveys, ethnographic and contextual enquiry
New Interactive Technologies	Choosing interaction styles and interaction techniques
	Approaches to design, implementation and evaluation of non-mouse interaction

TABLE V. LINGUISTICS FOUNDATION PER KUs' TOPICS IN INFORMATION MANAGEMENT KA

Knowledge Unit	***Topic***
Information Management Concepts	Information capture and representation
Query Languages	Overview of database languages
	Different ways to invoke non-procedural queries in conventional languages
Data Mining	Uses of data mining
	Data mining algorithms
Storage and Retrieval	Documents, electronic publishing, markup, and markup languages

TABLE VI. LINGUISTICS FOUNDATION PER KUs' TOPICS IN INTELLIGENCE SYSTEMS KA

Knowledge Unit	***Topic***
Basic Knowledge Representation and Reasoning	Review of propositional and predicate logic (cross-reference DS/Basic Logic)
Basic Machine Learning	Definition and examples of broad variety of machine learning tasks, including classification
Advanced Representation and Reasoning	Knowledge representation issues
Reasoning Under Uncertainty	Knowledge representations
Natural Language Processing (NLP)	Deterministic and stochastic grammars
	Parsing algorithms
	Representing meaning/Semantics
	Information extraction
	Language translation
	Text classification, categorization

TABLE VII. LINGUISTICS FOUNDATION PER KUs' TOPICS IN PLATFORM-BASED DEVELOPMENT KA

Knowledge Unit	***Topic***
Web Platforms	Web programming languages (e.g., HTML5, Java Script, PHP, CSS)
Mobile Platforms	Mobile programming languages
Industrial Platforms	Domain-specific languages
Game Platforms	Game platform languages

TABLE VIII. LINGUISTICS FOUNDATION PER KUs' TOPICS IN PROGRAMMING LANGUAGES KA

Knowledge Unit	***Topic***
Object-Oriented Programming (OOP)	Object-oriented design
Language Translation and Execution	Interpretation vs. compilation to native code vs. compilation to portable intermediate representation
	Language translation pipeline: parsing, optional type-checking, translation, linking, execution
Syntax Analysis	Scanning (lexical analysis) using regular expressions", "Parsing strategies including top-down (e.g., recursive descent, Earley parsing, or LL) and bottom-up (e.g., backtracking or LR) techniques; role of context-free grammars
	Generating scanners and parsers from declarative specifications

TABLE VIII. LINGUISTICS FOUNDATION PER KUs' TOPICS IN PROGRAMMING LANGUAGES KA

Knowledge Unit	*Topic*
Complier Semantic Analysis	High-level program representations such as abstract syntax trees
Advance Programming Constructs	Language support for checking assertions, invariants, and pre/post-conditions
Concurrency and Parallelism	Language support for data parallelism
Formal Semantics	Syntax vs. semantics
Language Pragmatics	Principles of language design such as orthogonality

TABLE IX. LINGUISTICS FOUNDATION PER KUs' TOPICS IN SOFTWARE DEVELOPMENT FUNDAMENTALS KA

Knowledge Unit	*Topic*
Fundamental Programming Concepts	Basic syntax and semantics of a higher-level language
Tools and Environments	Requirements analysis and design modeling tools
Requirements Engineering	Describing functional requirements using, for example, use cases or users stories
	Requirements analysis modeling techniques

In general, the linguistics is a direct foundational material for 9 Knowledge Areas, which represent the half of computer science Knowledge Areas.

IV. INCLUSION OF LINGUISTICS MATERIAL WITHIN KNOWLEDGE AREAS OF COMPUTER SCIENCE CURRICULA

The previous Section reviews the topics that the linguistics is a foundational material for them. This section views in details the linguistics foundational materials for the Knowledge Areas for those topics.

A. Intelligence System (IS) KA.

In Artificial Intelligence, linguistics is one of its foundational materials. Russell and Norvig make a similar point about the linguistics as foundational material for AI and the born of *computational linguistics* field:

In 1957, B. F. Skinner issued Verbal Conduct. This was an exhaustive and detailed recite of the behaviorist way to deal with language learning, composed by the first expert in the field. Inquisitively, the book's revise got as well known as the actual book, and worked to nearly end the interest in behaviorism. The owner of the revise was the linguist Noam Chomsky, who had recently issued a book on his own theorem, Syntactic Structures.

Chomsky brought up that the behaviorist theorem didn't process the idea of inventiveness in language—it didn't clarify how youngsters could comprehend and make up sentences that they had never heard. Chomsky's theorem—in light of syntactic models returning to the Indian linguist Panini (c. 350 BCE) — could clarify this, and not at all like past theorems, was it formal enough that it could initially be implemented. Recent linguistics and artificial intelligence, at that point, were "developed" at about a similar time, and grew up together, crossing in a half and half field called *computational linguistics* or *natural language processing*. It is clearly been that the issue of understanding language extensively more complicated than it appeared to be in 1957. Understanding language requires a comprehension of the topic and context, not simply a comprehension of the construction of sentences. This may appear clearly, yet it was not broadly appreciated until the 1960s. A significant part of the early work in knowledge representation (the investigation of how to place knowledge into a structure that a machine can thought with) was linked to language and educated by research in semantics, which was associated thus to many years of work on the philosophical examination of language. [3]

According to Sebesta, "Interest in AI showed up during the 1950s in various areas. A part of this interest outgrew from *linguistics*, some from psychology, and some from arithmetic. *Linguists* were worried about natural language processing. In addition to other basic functions of brain, Psychologists were keen on demonstrating human data stockpiling and recovery. Mathematicians were looked on automation of intelligent functions, for example, theory proving. These examinations come to a similar end result:

Some strategy should be created to permit computers to handle data in linked collections. At that point, most calculation was on numeric information in matrices." [5].

B. Programming Language (PL) KA.

According to Sebesta, "One of the most important events of 1959 was the presentation of the work of the Zurich committee to the International Conference on Information Processing, for there Backus presented his new documentation for depicting the syntax of programming languages, which later got known as BNF (Backus-Naur structure)… This effective utilization of the BNF formalism started a few significant fields of computer science: formal languages, parsing theorem, and BNF-based compiler building… It is noteworthy that BNF is almost indistinguishable from Chomsky's generative gadgets for context-free languages, called context-free grammar." [5].

Programmers use computer languages to write a program that directs a computer to perform specific task. Forouzan makes a similar point about the computer languages:

At the beginning of the computer age there was only one computer language, machine language. Programmers wrote instructions (using binary patterns) to solve a problem. However, as programs became larger, writing long programs using these patterns became tedious. Computer scientists came up with the idea of using symbols to represent binary patterns, just as people use symbols (words) for commands in daily life. Of course, the symbols used in daily life are different from those used in computers. So the concept of computer languages was born. A natural language such as English is rich and has many rules to combine words correctly: a computer language, on the other hand, has a more limited number of symbols and also a limited number of words. [4]

Like natural languages, programming language's syntax and semantics must be determined in order to implement it. According to Sebesta, "The consideration of programming languages, similar to the consideration of natural languages, can be splatted into investigations of syntax and semantics. The syntax of a programming language is the structure of its expressions, statements, and modules. Its semantics is the significance of those articulations, explanations, and program units." [5].

Lisp programming language was appeared due to the concern of *linguists* with natural language processing [5]. Natural Language Processing (NLP) is one of major AI branches. The conjunction of AI tool with *linguistics* advance the scientific understanding of languages and language use [3].

Finally, Russell and Norvig state, "Work on implementation of language processing is introduced at the biennial Applied Natural Language Processing conference (ANLP), the conference on Empirical Methods in Natural Language Processing (EMNLP), and the journal Natural Language Engineering. A broad range of NLP work appears in the journal *Computational Linguistics* and its conference, ACL, and in the *International Computational Linguistics* (COLING) conference." [3].

C. Human Computer Interaction (HCI) KA.

The user's interaction with computer is based on *linguistic*. "The human's interactivity with a machine is regularly seen on the one hand of language, so it isn't amazing that many modeling formalisms have created revolved around this idea. Many of the discussion visualization tools depicted in Section 16 are likewise founded on linguistics notion. In fact, BNF grammars are regularly used to determine discussions. The models here, albeit comparable in structure to discussion visualization tools, have been proposed determined to comprehend the human's conduct and understanding the cognitive difficulty of the interface." [8].

Linguistics is a one of foundational materials of HCI. MacKenzie makes similar point about linguistics as foundation of HCI:

HCI itself doesn't feel "narrowly concentration." Oppositely, HCI is enormously expansive in scope. It depends on concerns and practice in disciplines like psychology (especially cognitive psychology and experimental psychology), sociology, anthropology, cognitive science, computer science, and linguistics. [7]

Brain is one of descriptive models for the human that "assist us with beginning in understanding the human, to specify and classify parts of the human that are pertinent to HCI" [7]. Cognition is one of brain faculties"— the human operation of conscious intellectual activity, like reasoning, thinking, and choosing. Cognition extends numerous fields—from neurology to linguistics to anthropology—and, as anyone might expect, there are contending sees on the range of perception" [7].

Naturalness is one of HCI principles, and it alludes to "a characteristic that is mirrored different processes in our regular daily existence. For example, an ideal HCI may one day be acknowledged when a natural language–based conversational interface is conceivable, on the grounds that this is the common way that people communicate." [6]. Natural-language is one of major user-interface input methods, and the service of understanding the natural-language is the future of human-computer interaction. According to Kim, "Machine understanding of long clauses and natural-language-depend orders is still computationally troublesome and requesting. While not quite practical for everyday user-interface input methods, language-understanding technology is advancing fast, as demonstrated recently by the Apple® Siri [15] and IBM® Watson [16], where high-quality natural-language-understanding services are offered by the cloud (Figure 3.14). Captured segments of voice/text-input sentences can be sent to these cloud servers for very fast and near-real-time response. With the spread of smart-media client devices that might be computationally light yet equipped with a sleuth of sensors, such a cloud-based natural-language interaction (combined with intelligence) will revolutionize the way we interact with computers in the near future" [6].

D. Computational Science (CN) Knowledge Area.

The Theory of Computation field deals with subjects that "look up for specifying what can and can't be processed, how rapidly, with how much memory, and on which sort of computational model. [9]"

Automata theory is a one of the central areas of theory of computation. According to Sipser, "Automata theorem addresses the definitions and properties of mathematical models of computation. These models function in many applied fields of computer science. One model, called the *finite automaton*, is utilized in text handling, compilers, and device design. Another model, called the *context-free grammar*, is utilized in *programming languages* and *AI*." [9].

Context-free grammar is a one of methods that are used to describe languages, which "were first used in the study *human language*… A significant utilization of context-free grammars happens in the description and implementation of programming languages. A grammar for a programming language frequently shows up as a reference for individuals attempting to study the language syntax." [9].

Computer scientists benefited from linguists by using natural language definition tools to define programming languages. Sudkamp makes similar point about context-free grammar to define programming language:

Formal language theory was initiated by question, "How are languages defined?" in an attempt to capture the structure and nuances of natural languages, linguist Noam Chomsky developed formal systems called *grammar* for defining and generating correct sentences. At approximately the same time, computer scientists were grappling with the problem of explicitly and unambiguously defining the syntax of programming languages. These two examinations merged when the syntax of the programming language ALGOL was defined using a formalism equivalent to a context-free grammar. [10].

E. Software Development Fundamentals (SDF) and Platform-Based Development (PBD) Knowledge Areas.

In software engineering, the software development processes consist of several activities, actions, and tasks. Natural language plays critical roles in performing such activities, actions, and tasks. Different languages are designed to implement different applications in different development platforms such as web and mobile. As explained in previous sections, the linguistics is foundational material for programming languages, which in turns the foundational material for platform-based development.

Regards to SDF, "the formal methods used in developing computer systems are mathematically based methods for modeling system properties… The formal syntax of a specification language (Appendix 3) allows requirements or design to be translated in only one way, removing fuzzy that frequently happens when a natural language (e.g., English) or a graphical visualization tool (e.g., UML) should be interpreted by a reader… A formal specification language is typically made out of three essential segments: (1) a syntax that determines the particular notation with which the specification is represented, (2) semantic domain to help identify a "world of objects" [Win90] that will be utilized to depict the system, and (3) a group of relations that identify the semantic standards that show how objects might be handled appropriately and fulfill the specification" [11]. As the previous section mentioned, language are implemented based on linguistic ideas.

On the other side, AI techniques (such as data mining, machine learning, and neural networks, which in turn based on linguistics ideas) support software engineering [12].

Grammar as linguistics tool is used in design task in software development. Sommerville states, "As object-oriented design evolved in the 1980s, various ways of identifying object classes in object-oriented systems were suggested: 1. Use a grammatical analysis of a natural language description of the system to be constructed. Objects and attributes are nouns; operations or services are verbs (Abbott 1983)" [13].

F.Information Management (IM) Knowledge Area.

"Practically speaking, the two essential objectives of data mining are *prediction* and *description*. Prediction involves using some variables or fields in the data set to predict unknown or future values of other variables of interest. Prediction includes utilizing some variables or fields in the data set to foresee obscure or future upsides of different factors of interest. Description, on the other side, centers on discovering types that describing data that can be translated by people.… These ambiguous sets, which generally holds names that adjust to adjectives showing up in our everyday semantic use, for example, "huge," "medium," or "little," are called *linguistic* values or *linguistic* labels. Subsequently, the universe of talk X is frequently called the linguistic variable… The *linguistic* representation drives to the disclosure of natural and more understandable." [14]. Consider the last quote, the linguistics ideas obviously appear in data mining field.

V. CONCULTION

The paper explores the balance of linguistics as a foundational material in computer science and its application. The linguistic is a foundational material for at least half of computer science knowledge areas. This gives the importance of linguistics to both students and specialists in computer science, which is no less than the mathematics and others foundations. Recognizing and understanding of required linguistics foundation help students and researcher in their real contributions in computer science discipline.

ACKNOWLEDEMENT

The author thanks his colleague Dr. A. Hamzah for his helping in reviewing this paper.

REFERENCES

[1] ACM Computing Curricula Task Force, Ed., Computer Science Curricula 2013*: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, Inc., 2013.

[2] PROJECT MANAGEMENT INSTITUTE., *Guide to the project management body of knowledge*. [s.l.]: Project Management Inst., 2018.

[3] S. Russell and P. Norvig, *Artificial intelligence: A modern approach*, 4th ed. Upper Saddle River, NJ: Pearson, 2020.

[4] B. Forouzan, *Foundations of Computer Science*, 4th ed. Andover, England: Cengage Learning EMEA, 2017.

[5] R. W. Sebesta, *Concepts of programming languages*, 11th ed. Upper Saddle River, NJ: Pearson, 2015.

[6] G. J. Kim, *Human-computer interaction: Fundamentals and practice*. Philadelphia, PA: Auerbach, 2015.

[7] I. S. MacKenzie, *Human-computer interaction: An empirical research perspective*. Oxford, England: Morgan Kaufmann, 2012.

[8] J. E. Finlay, A. Dix, R. Beale, and G. D. Abowd, *Human-Computer Interaction*, 3rd ed. Philadelphia, PA: Prentice Hall, 2003.

[9] M. Sipser, *Introduction to the theory of computation*, international edition, 3rd ed. Florence, AL: South-Western College Publishing, 2012.

[10] T. A. Sudkamp, *Languages and machines: An introduction to the theory of computer science*: United States edition, 3rd ed. Upper Saddle River, NJ: Pearson, 2005.

[11] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th Ed. New York, NY: McGraw-Hill Professional, 2014.

[12] D. P. Wangoo, "Artificial Intelligence Techniques in Software Engineering for Automated Software Reuse and Design," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, 2018.

[13] I. Sommerville, Software engineering, 10th ed. Upper Saddle River, NJ: Pearson, 2016.

[14] M. Kantardzic, Data mining : concepts, models, methods, and algorithms. New York: Wiley-Interscience, 2015.