

A Concept Map based Teaching of Compiler Design for Undergraduate Students

Venkatesan Subramanian^{1,*}, Kalaivany Karthikeyan², Pallapa Venkataram³

¹Department of Information Technology, Indian Institute of Information Technology, Allahabad, India.

²Independent Researcher, Australia.

³Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India.

Abstract

In undergraduate engineering, most of the subjects do not have the open visibility of the Industry and Research requirements. Students are interested mostly on subjects which are useful for Industry placement. They do not show interest in non-open visibility subjects if an instructor teaches by simply following the textbook. Considering this, we presented a concept map based teaching methodology with Research and Industry assignments and problems. The proposed methodology focus on improving the teaching quality and students understanding level. In this paper, we have taken the Compiler Design subject and presented the concept map. To understand the effectiveness of the proposed methodology, the students feedback was collected and evaluated using the sign-test and the students' submitted problems and assignments were evaluated to understand their level. The analysis results show that most of students studied Compiler Design with interest as a result of proposed teaching methodology.

Received on 21 September 2022; accepted on 24 July 2022; published on 17 August 2022

Keywords: Compiler Design, Concept Map, Computer Science education, Teaching

Copyright © 2022 Venkatesan *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eetel.v8i1.2550

1. Introduction

The Industry professionals and the Academicians claim that most of undergraduate engineering students are not having required potential for hiring. The major reason behind such criticism is that while studying the course, students do not show interest in the subjects, thus affect their understanding of the subject in depth and they were not motivated to show interest in the subjects. Also, M. J. De Vries and J. H. M. Stroeken [1] found that engineering students are lacking in research skills (the abilities to define the research problem, comment upon research methodology, and reflect upon research outcomes). This varies from one subject to another subject. To understand the students' thoughts about different subjects, a survey was conducted with the three questions.

- What are the subjects you are interested in?

- What are the subjects you feel are required for Industry placement?
- What are the subjects you feel are required for Research?

The third-year undergraduate engineering students (number of students participated in the survey is 92) participated in the survey and given their opinion. The survey results in percentage are shown in Table 1 for few subjects of undergraduate computer science and engineering. The same survey was repeated with the next batch of third year engineering undergraduate students (number of student participated in the survey: 99) with a following additional question.

- If the listed subjects are not given in a core list but offered as elective, then what are the subjects will you prefer to study.

The survey result in percentage is shown in Table 2. It is very clear from the survey results that students are not having enough interest on some subjects as well as not aware of its need in Industry and Research. Hence,

*Corresponding author. Email: venkat@iita.ac.in

Table 1. Students opinion about subjects - Batch I (2019)

Subject	Interested	Placement	Research
Compiler Design	14	14	19
Computer Networks	37	52	9
Programming	45	52	2
Data Structure	81	95	17

Table 2. Students opinion about subjects - Batch II (2020)

Subject	Interested	Placement	Research	Elective
Compiler Design	10	12	26	16
Computer Networks	48	61	17	39
Programming	51	61	14	37
Data Structure	92	96	35	71

it is necessary to encourage and motivate students toward these subjects also by changing the teaching methodology, which should be in a high degree of quality. The concept map [2] [3] based teaching would be more appropriate for the purpose as C. Chiou et al. [4] presented its effectiveness.

In this paper, Compiler Design subject is concentrated since students feel this is a complex and very less productive. The students' opinion in Table 1 and 2 clearly shows that the students have a negative notion about the Compiler Design subject. Hence, Instructor need to beat this notion by re-designing the teaching methodology of Compiler Design and motivating students towards the subject.

1.1. Research Questions

While teaching, an instructor plays a significant role in motivating and making students to understand the subject. This can be achieved if an instructor have solutions to the following questions.

- Question 1: What innovative technique should be introduced to beat the students' negative notions of various subjects? - Most of the students in the class are not interested in the subjects that are not directly visible as a requirement for industry and research demand. This can be witnessed from the Table 1 and 2. The Instructor needs to follow an innovative strategy to beat the notion.
- Question 2: How do instructor make students self-design the problem and assignments for a concept of a subject? - The students in a subject understand the concept when the Instructor teaches it or later in a few cases. Also they can solve the problem given in the textbook. In case

the problem is tricky, then they are unable to solve it. The instructor can overcome this issue by asking the students to design problems and assignments independently. For example, Ana et al. [5] proposed a module using concept map to support self-regulated learning by undergraduate engineering students. Each student are linked to the concept map to record all topics that are covered and the ones that are pending. The Instructor examines the concept maps of all students for further improvement. However, the issue is what will encourage the students to do it?

- Question 3: What factor is required to make the students in the classroom to be attentive? - In most of cases, the students are not attentive in the classroom due to various factors such as teaching methodology, complex topics, smart gadgets, next guy nuisance, etc. In a classroom of more than 50 students, the instructors may not be in a position to monitor all students. Even if the instructor forces or threatens the students to listen to the teaching, it will be a physical presence but mental absence which is non productive. Hence a novel teaching technique is required to make the students be attentive in the classroom.

The significance of the above questions are to achieve the long-lasting remembrance of the subjects, properly utilize the learned concepts in real time on demand, and most importantly, make the full-fledged graduate. Considering the significance of the questions, benefits of concept map and students opinion, the novel teaching methodology is introduced with an evaluation technique to provide solutions to the research questions. The methodology has been

experimented while teaching the Compiler Design subject for Undergraduate engineering students.

1.2. Contribution

The contributions of the paper to improve the teaching quality and understanding level of students are

1. A novel teaching methodology based on the concept map, directed concept graph, and concept relation weight.
2. Linked each concept with the present and future Industry and Research related problems and assignments.
3. Developed the broad and extended concept map of Compiler Design.

The rest of the paper is structured as follows: Section 2 discusses the existing works, Section 3 discusses the proposed teaching research methodology, section 4 discusses the broad and extended concept map of the Compiler Design subject. Section 5 presents the evaluation of the proposed methodology and section 6 proves the significance of the proposed teaching methodology and discusses the limitation. Finally, Section 7 concludes the paper with the directions of future work.

2. Related works

In literature, different approaches were proposed to encourage and motivate the students to concentrate on Compiler Design subject and its concepts. Here, we discuss the existing works in two categories: The Teaching style of the compiler design course and its concepts.

H. Liu [6] introduced software engineering practice in an Undergraduate Compiler Course. A. Demaille et al. [7] introduced a set of compiler construction tools for educational projects. L. Xu and F. G. Martin [8] proposed a Chirp system that provides a realistic and engaging environment to teach compiler courses. T. R. Henry [9] proposed a Game Programming Language (GPL) based teaching to motivate students towards the compiler project. E. White et al. [10] proposed an approach that enables students of compiler courses to examine and experiment with a real compiler without becoming overwhelmed by complexity. S. R. Vegdahl [11] proposed visualization tool to teach compiler design to bring the interest of students to the subject. M. Mernik and V. Zumer [12] described a software tool called LISA (Language Implementation System Based on Attribute Grammars) for learning and conceptual understanding of compiler construction in an efficient, direct, and long-lasting way. H. D. Shapiro and M. D. Mickunas [13] replaced the term project

on compiler design with several smaller, independent programming assignments to better understand and motivate students. M. Ruckert [14] argues that teaching compiler with unusual programming language is a good choice. D. Kundra and A. Sureka [15] [16] discussed case-based teaching and learning of compiler design and proposed case studies for different concepts to make learning easier and more interesting. N. Wang and L. Li [17] reformed the compiler theory course by combining theory and practice to improve the students' skill set.

J. Velásquez [18] developed a tool called bcc: mini-language to cover the complete syllabus in a semester. The bcc supports to execute different phases of the compiler. K. Abe [19] developed an integrated laboratory that deals with processor organization, compiler design, and computer networking. The students are encouraged to integrate these components to construct a complete small computer system. J. Velásquez [20] also described exposing students to the practical application of compiler construction techniques and encouraging the development of programming and problem-solving skills. In addition, the author discussed the implementation of a scheme for the automatic assessment using the Virtual Programming Lab. S. B. Nolen and M. D. Koretsky [21] discussed the physical projects to actively engage students in a subject. M. Frank et al. [22] analysed and recommended teaching technology and engineering by means of Project Based learning can train students better for their future profession. J. Àngel Velázquez-Iturbide et al. [23] analyzed and identified that the student motivation increase through software visualization. I. Estévez-Ayres et al. [24] proposed a methodology for improving active learning through regular feedback and iterative refinement.

F. Ortin et al. [25] shows how object-oriented design patterns represented in unified modeling language (UML) can be used to teach type checking and develop the semantic analysis phase of a compiler. S. Sangal et al. [26] proposed a Parsing Algorithms Visualization Tool (PAVT) to teach the process of parsing algorithms. Learners can visualise the intermediate steps of the parsing algorithm through the tool. R. D. Resler and D. M. Deaver [27] introduced the Visible Compiler Compiler (VCOCO) to make the students visualize a compiler's internal working. The VCOCO is flexible and effective tool for generating user-specified LL(1) visible compilers. A. Karkare and N. Agrawal [28] developed ParseIT tool for teaching parsing.

The game based, experiment based, project based like teaching computer architecture by D. Cenk Erdil [29], case based, feedback based and tool based teaching makes learning easy and interesting. However, concepts and its relationship along with assignments and problems related to Research and Industry requirement

should be discussed for motivation and long-lasting remembrance thus provides solutions to the research questions. Hence this paper propose a novel teaching methodology based on concept map which will simplify the teaching, make students to understand the subject and get motivated.

3. Research Methodology

This section discusses the proposed novel teaching research methodology and its evaluation technique to answer the teaching research questions to improve the quality of teaching and understanding.

3.1. Methodology

The concept map based teaching and relating the concepts with Research and Industry demand is the right method to address the above research questions since students are interested in Industry which was witnessed through the survey outcome. Considering this, a systematic teaching methodology is introduced to teach a subject.

Basics for Quality Teaching. For efficient and qualitative teaching, a teacher/instructor shall follow the research questions and plan the subject's teaching material, assignments and problems. This paper improves the quality of teaching by the teacher by constructing the concept map and directed graph of the subject, which gives a teacher to systematically follow the relation of concepts and order to teach with examples. Also, to evaluate the students' understanding and improve their understanding level, students needs to self-design the problems and assignments. Accordingly, the following two teaching experiments are designed.

Teaching quality improvement The systematic approach of teaching is to identify, order and deliver the concepts. While teaching, each concepts should be discussed along with practical Industry and Research usage. Also, the relevant assignments and problems of a concept to be discussed before taking up the next concept in the order or queue. It is necessary to make the queue and follow it so that students will not miss the significance of any concepts and their relational concepts. The Directed Concept Graph ($\mathcal{G} = (\mathcal{N}, \mathcal{E})$, the set of N nodes and \mathcal{E} edges) is one of the best techniques to order the concepts of a subject. The nodes \mathcal{N} are the concepts, and edges \mathcal{E} are the concepts' relations. The Figure 1 shows the Directed Concept Graph (DCG) with five concepts notation C_a, C_b, C_c, C_d and C_e , which can be mapped with compiler concepts.

The Lemma 1 proves the need for DCG to order the concepts of a subject and discuss them in the class, thus improving the teaching quality.

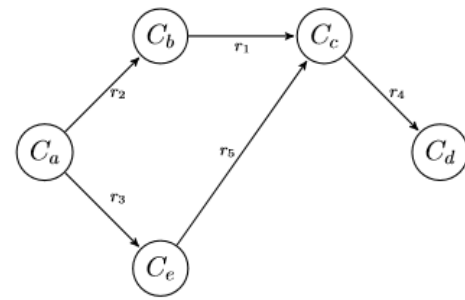


Figure 1. Directed Concept Graph

Postulate 1. A concept's pre-requisite concept will have the significant influence on understanding it.

Postulate 2. A source node (concept) C_a , is the pre-requisite for the destination node (concept) C_b , if the directed relationship is from $C_a \rightarrow C_b$

Postulate 3. A concept should be discussed only after its pre-requisite concepts are discussed with relevant problems and assignments.

Lemma 1. The concept queue get ordered concepts if the DCG followed.

Proof. Given postulate 1, 2 and 3, the precedence of concept C_a is more than concept C_b ($C_a > C_b$), if the C_b has directed edge from the C_a . Therefore, the C_a comes before C_b in the concept's queue. Hence the queue will have ordered concepts. \square

The directed edge is insufficient to order the concepts when a concept has more than one outdegree. For example, C_a has directed edge (two outdegrees) to C_b and C_e . The next concept to C_a in the order is non-deterministic. To overcome the non-determinism, the relation weight among concepts is introduced. An each concept of a subject will be given weight based on the importance in three categories; teaching (use in other subjects), industry and research. The qualitative weight under each category is classified into low, medium and high with a quantity of 1, 2 and 3 respectively. A concept's overall or total weight is the summation of weights of all three categories. The weights for some concepts of the Compiler Design is given in Table 3. The weights are assigned based on the Instructors' experience in teaching, preparation and evaluation of the Industry and Research problems and assignments.

Using the concept weights, the relation weight can be assigned. For example, the relation weight of $C_a \rightarrow C_b$ is sum of weight of C_a and C_b . The relation weight can be used to decide the next concept to be placed in the order.

What will be the loss if the instructor miss a concept?: A concept missed while teaching a subject will greatly impact on understanding the related concepts. For

Table 3. Concept Weight

Concept	Teaching	Industry	Research	Sum
Regular Expression	3	3	3	9
Optimization	3	3	3	9
Attribute Grammar	2	1	2	5
Finite State Machine	3	3	3	9
Left Factoring	1	1	1	3
Three Address Code	2	2	2	6

example, an instructor teaching LR parsing to the students should teach the item set construction, parsing table construction and parsing in order. Lets assume the Instructor miss the item set and parsing table construction but teaches parsing to the students. In this case, students will keep on thinking about how the table was constructed and what are its input. This forces students on the dark side, thus, loose interest in the concept and eventually loose interest in the subject.

What will be the loss if the Instructor does not follow the order?: A concept will significantly influence understanding the other concepts. In case, the concept C_a , which should be discussed before concept C_b is missed or discussed later, then the student will not be able to understand C_b . For example, lets map the DCG of Figure 1 with the lexical analyser phase concepts of Compiler Design. The C_a is the Tokenization, C_b is the Finite State Machine, C_c is the Regular Expression, and C_d is the Table Driven LA. Assume, the instructor teaches C_a, C_b, C_c in order but the C_d later. In this case, students will not be in position to understand the source of C_d while it is discussed thus degrade the teaching quality. This supports the need to order concepts using DCG and relation weight to improve the teaching quality.

Improving understanding level of students In addition to the concept map and order-based teaching, the pyramid structure, as shown in Figure 2 can be followed while teaching each concept. An idea is to give the definition, practical working model, how it was and is used and what would be its future. The structure shows the contents of a concept to be discussed and the time frame for each content. The time frame is proportional to spread of the pyramid. The discussion of lexical analyzer concept based on the pyramid shown in Figure 2 is as follows.

Firstly, the concept was defined then its working model along with text book based examples were discussed. Later, use of the concept in an other concepts or subjects was discussed. At the end, the research and industry oriented examples of the concept were discussed by the Instructors. Finally, students were

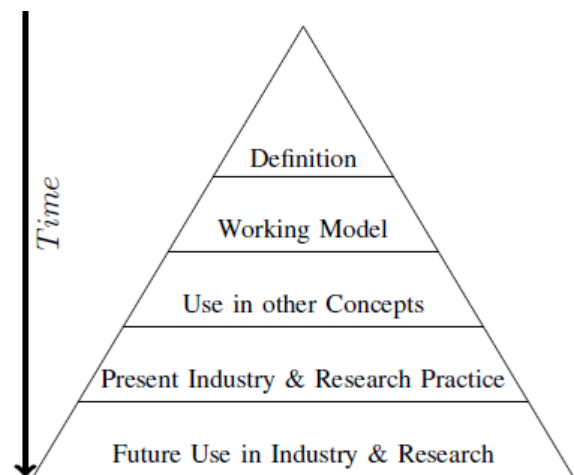


Figure 2. Pyramid based Concept Teaching Method

invited for the discussion to describe the future need of the concept in Industry as well as in the research domain. The student must take up the concept only when its pre-requisite concepts are clear.

Need of pre-requisite concept understanding: If a student does not have 100% understanding of C_c of Figure 1 then cannot solve the problem of C_d . For example, a student not having clear understanding of *first* and *follow* construction cannot solve the example problem or prepare the sample assignment or problem for the item set/parsing table construction. Hence, understanding the pre-requisite concepts by the students is very essential and instructor needs to guide them.

To understand the concept very clearly, the students should be asked to self-design Industry, and Research problems for each concepts based on the Instructor discussed Industry and Research examples. A student with sufficient knowledge on the concept can prepare the relevant industry and research assignments and problems. The instructors can understand students' level of understanding while evaluating their assignments and problems accordingly, which can guide them to understand the concept in depth. This will make the students think innovative and attentive in the classroom. Also,

this method will encourage the students towards the subject since they visualize and realize the use of a concept in Industry and Research domain. Hence, this method improves the understanding level of students.

Building Concept Map. The subject overview can be presented using a concept map to encourage students toward a subject. In the experiment, the Compiler Design concept map was built and discussed in the class. The process of making the broad concept map is presented in the Algorithm 1. As a part of building the concept map, each concept must be linked with other concepts if there is an association/relation, as discussed in the Directed Concept Graph. The associations/relationships are has, includes, creates, etc. The extended concept map was built for each phase of compiler based on the concepts of broad concept map and added supplementary such as what it is?, How does it works?, Objective, Implementation, Approaches to build, What it use?, industry and research example, etc. To differentiate the concepts and supplementary, separate representation was used. The red color border rectangle is used for supplementary in the extended concept map of Compiler Design discussed in section 4.2.

Algorithm 1 Building Broad Concept Map

Require: List of Concepts $\mathcal{C} \subset \{c_1, c_2, c_3..c_n\}$

Ensure: Concept Map \mathcal{CM}

```

1: for  $c_i$  in  $\mathcal{C}$  do
2:   Design a simple example for  $c_i$  based on text book

3:   if  $c_i$  is the major concept with weight  $\geq 1$  in
   Industry and Research then
4:     Identify the Research and Industry related
   assignments
5:   end if
6:   for  $c_j \{c_1, c_2, c_3..c_{i-1}\}$  in  $\mathcal{C}$  do
7:     if  $c_j$  has relationship with  $c_i$  then
8:       Identify the concept map relationship  $\mathcal{R} \subset$ 
       {has, includes, creates, can be,... }
9:       Build the relationship  $\mathcal{R} : c_j \rightarrow c_i$ 
10:    end if
11:  end for
12: end for
13: return  $\mathcal{CM}$ 

```

3.2. Evaluation Technique

The proposed methodology can be evaluated in two ways: the first is the evaluation of the proposed teaching methodology and second is the evaluation of students' self-designed assignments and problems.

Methodology Evaluation. The proposed teaching methodology can be evaluated using the students'

feedback survey. The survey can be conducted at beginning and at the end of the course with the same set of students and questions. The course should be taught using the proposed teaching methodology. The survey questions to be prepared to understand the students' mind set about a subject and the response to be dichotomous, that is *yes* or *no*. In section 1, the necessary survey questions are presented to understand the students, mindset about a subject. The outcome of the survey results can be evaluated using the non-parametric sign test [30] to prove the significance of the proposed methodology. The *sign test* uses the binomial distribution with the cumulative distribution function given in equation 1.

$$F(k, n, p) = Pr(X \leq k) = \sum_{i=0}^k \binom{n}{i} p^i (1-p)^{n-i} \quad (1)$$

The *sign test* is most useful if comparisons can be expressed as $x > y$, $x = y$ and $x < y$ [31], where x and y are the survey outcome before and after the event respectively. This fits the proposed methodology evaluation. Here, the event refers to the proposed teaching methodology. The possible outcome from the survey results x and y that can be used for *sign test* are as follows

- If the survey outcome y of a student has more answers as *yes* then record a '+' (positive for the proposed teaching methodology), where $x < y$.
- If the survey outcome x of a student has more answers as *yes* then record a '-' (negative for the proposed teaching methodology), where $x > y$.
- If a student's number of *yes* answers has not changed then record a '0' (neutral), where $x = y$.
- If a student's number of *no* answers has not changed, then record a '-' (negative for the proposed teaching methodology), where $\neg x = \neg y$.

To compute the binomial probability (bp) for *sign test*, n , total count that is number of '+' and '-' excluding '0', k , number of '-' that is against the proposed methodology and p , initial non biased probability value 0.5 are required. The two-tailed *sign test* is required since not sure whether *yes* answers would increase or decrease after the event. The hypotheses for the test are as follows.

- The null hypothesis H_0 : The proposed teaching methodology does not have any effect or a negative effect on the students.
- The alternate hypothesis H_a : The proposed methodology has the positive effect on the students, and they are motivated.

Student Assignment Evaluation. The students submitted self-designed assignments and problems can be evaluated by Instructors according to the concept in the concept map and categorize into relative/non-relative or strong/weak. This helps the Instructors understand student's understanding level and their involvement. The non-relative or weak assignment submitted students could be asked to prepare the new set of assignments. Instructors can provide support in preparing the assignments by giving more examples.

4. COMPILER DESIGN CONCEPT MAP

The students studied or studying Compiler Design course should know the overview and understand how the compiler is designed, and the high-level program is getting translated for execution. The compiler design concept map [32] in Figure 3 provides the complete overview and relationship among the concepts to encourage the students to focus on the subject. The concept map includes the pre-processor and pre-requisites for this subject. The pre-requisites are instruction set, Context Free Grammar (CFG) and Context Sensitive Grammar (CSG), Regular Expression (RE), Finite State Machine (FSM) and Push Down Automata (PDA). This makes students understand the importance of Theory of Computation (ToC) subject since they studied RE, FSM, PDA and CFG/CSG there. Hence, the proposed concept map motivates the students toward Compiler Design and Theory of Computation subject. However, no advantage concerning ToC since students already studied the subject, but this can impact junior students because senior students will convey the importance of ToC/Automata to juniors.

4.1. Phases of Compiler

In this paper, machine independent optimization is included in the code optimization phase and machine dependent optimization in the code generation and optimization phase.

Lexical Analysis. The important concepts to be discussed in lexical analysis phase are Tokenization, Finite State Machine, Regular Expression. In addition, buffering and Symbol Table and obviously how these concepts are interlinked to generate the tokens. Even though, students studied regular expression and finite state machine in the pre-requisite subject ToC, the concepts need to be revised with programming language patterns example. In the classroom, instructor may take the hello world C language program as input and recognize the tokens using single and double buffering to better understand lexical analyser. The tools Flex, and lolo need to be briefly introduced and practiced in the laboratory.

Syntax Analysis. Syntax Analysis takes the input as tokens from the lexical phase and produces the syntax tree, which the semantic phase will use. The major task of the syntax analyzer is to check whether the syntax presents in the program is part of the programming language or not. To do this check, Syntax Analyzer uses parser with Finite State Machine, Push down Automata, Context Free and Sensitive Grammar. Parser can be top down or bottom up approach and it will be chosen based on the developer's requirement. The LL (Left-to-right, Leftmost derivation) and LR (Left-to-right, Rightmost derivation) parsers work well only when a grammar is unambiguous and parse in linear time. For LL parser, an unambiguous, deterministic and non-left recursive grammar will be given as input and compute the *first* and *follow*. Using the *first* and *follow*, the parsing table will be constructed to do parsing. In the case of the LR parser, the finite state machine based item set and parsing table for the unambiguous grammar will be constructed to do parsing. Instructor also needs to briefly discuss about the universal parsers like Earley's and CYK, which can take any type of grammar and parse however the complexity will not be linear. The tool Yacc can be briefly introduced. In Compiler Design course teaching, at least 25% to 35% of time in the semester will be spent on the syntax analysis phase.

Semantic Analysis. It takes the parse tree as input from the syntax analyzer and checks the semantics of the program such as type checking, parameter matching, label check, etc. This phase uses the attribute grammar or direct method to verify the semantics. The concepts in the semantic analyser are the Symbol Table, Attribute Grammars, Semantic check, Syntax Directed Translation and Definition. Attribute grammars can be represented using the Syntax Directed Translation (SDT) or Syntax Directed Definition (SDD) with Left to Right L- or Synthesized S-Attributed and use Dependency Graph at the time of evaluation to maintain the order. The semantic analyzer utilizes the symbol table to check the semantics of the program. There are high level programming languages do not have the attribute grammars for semantic check and it uses the symbol table.

Intermediate Code Generation (ICG). It takes the input as syntax tree from the semantic phase and provides the Intermediate Representation (IR). The IR can be structural, linear and hybrid. In most of the programming languages, linear representation is used further three address code is mostly preferred with storage representations such as quadruple, triple and indirect triple. It also uses the Single Static Assignment (SSA) for better register allocation. This phase also uses the SDT in some languages to convert the syntax tree representation into three address code.

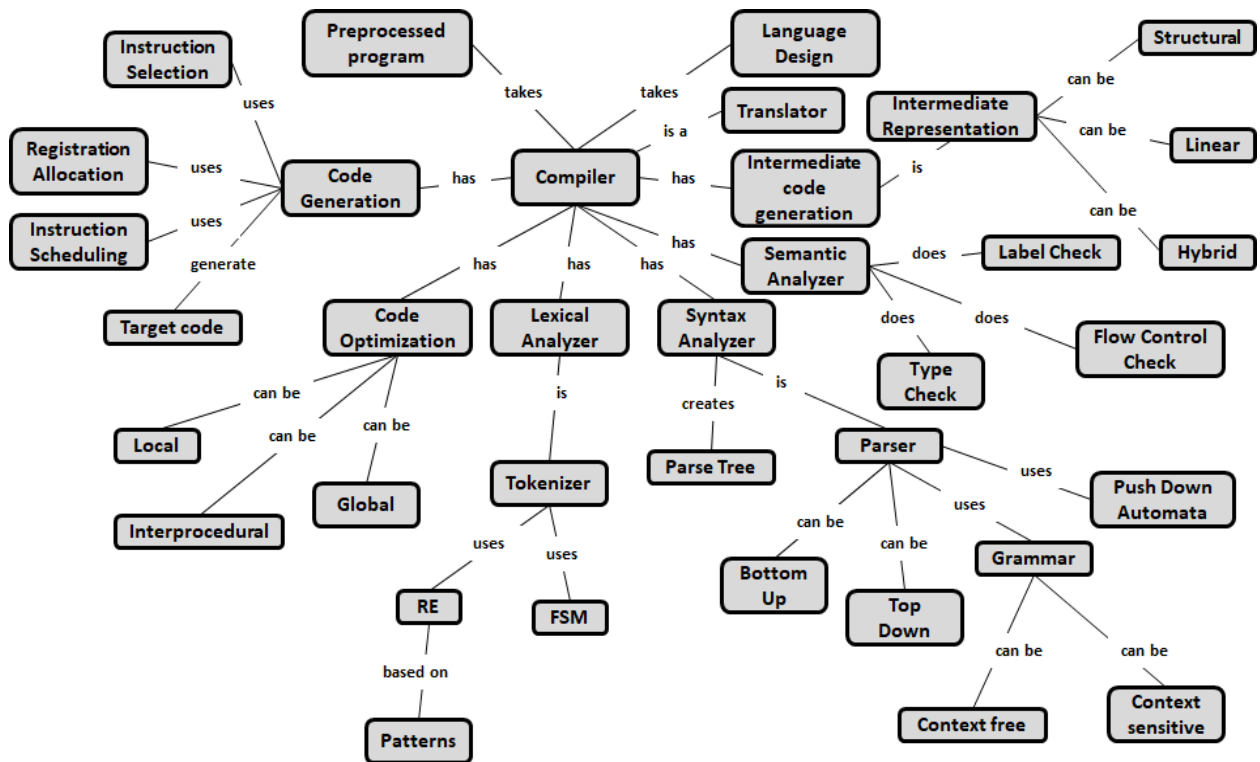


Figure 3. Basic concept map of Compiler Design

Code Optimization. This phase helps in optimizing the time and space complexity of the program while execution however semantics of the program should be preserved. The Code optimization can be done locally, globally or inter-procedural level and can also be machine dependent or independent. In the classroom, Instructor has to show one example for each optimization concept to make the concepts clear to students.

Code Generation. Code Generation is the final phase of the compiler that takes optimized code and generates target code. Target code may be another High level language code or Assembly level language code. To generate the assembly level, it is important to consider the Instruction Set, Register Allocation and Instruction Scheduling. Instruction set will be based on Reduced Instruction Set Computer (RISC) or Complex Instruction Set Computer (CISC) or Micro-op (mix of CISC and RISC). Since CISC architecture is used in popular Intel processors, CISC instruction set can be taken for solving the example in classroom. However, students may be asked to generate the target code for different architectures.

4.2. Extended Concept Map

The Instructor introduced the broad concept map at the first level then for each concept, the concept map

is extended further till it makes the students to understand easily and solve the problems and assignments. The extended concept map also encourages the enthusiastic student to design their problems and assignment regarding Industry application and Research. This section, presents the extended concept map of Compiler Design phases.

Lexical Analysis. The Figure 4 shows the extended concept map of lexical analyser [33][34]. The Industry and Research example for the core concepts in the extended concept map are as follows.

- 1I: Use the regular expression for medical industry to identify the external analytic vendor that is different from the approved list.
- 1R: Use text manipulation (update, delete, search) in the text file database using the regular expression [35]. In this case, flat files may play a useful role as a database. The extensive facilities provided by the databases such as MySQL, Oracle, SQL server, etc. are not required and free the users from the additional work of database software installation, knowing a query language etc.
- 2I: Finite State Machine (FSM) can be used to execute sequence of tests for flow measurement testing. This solution can be applied to automate other serial, and batch processes [36].

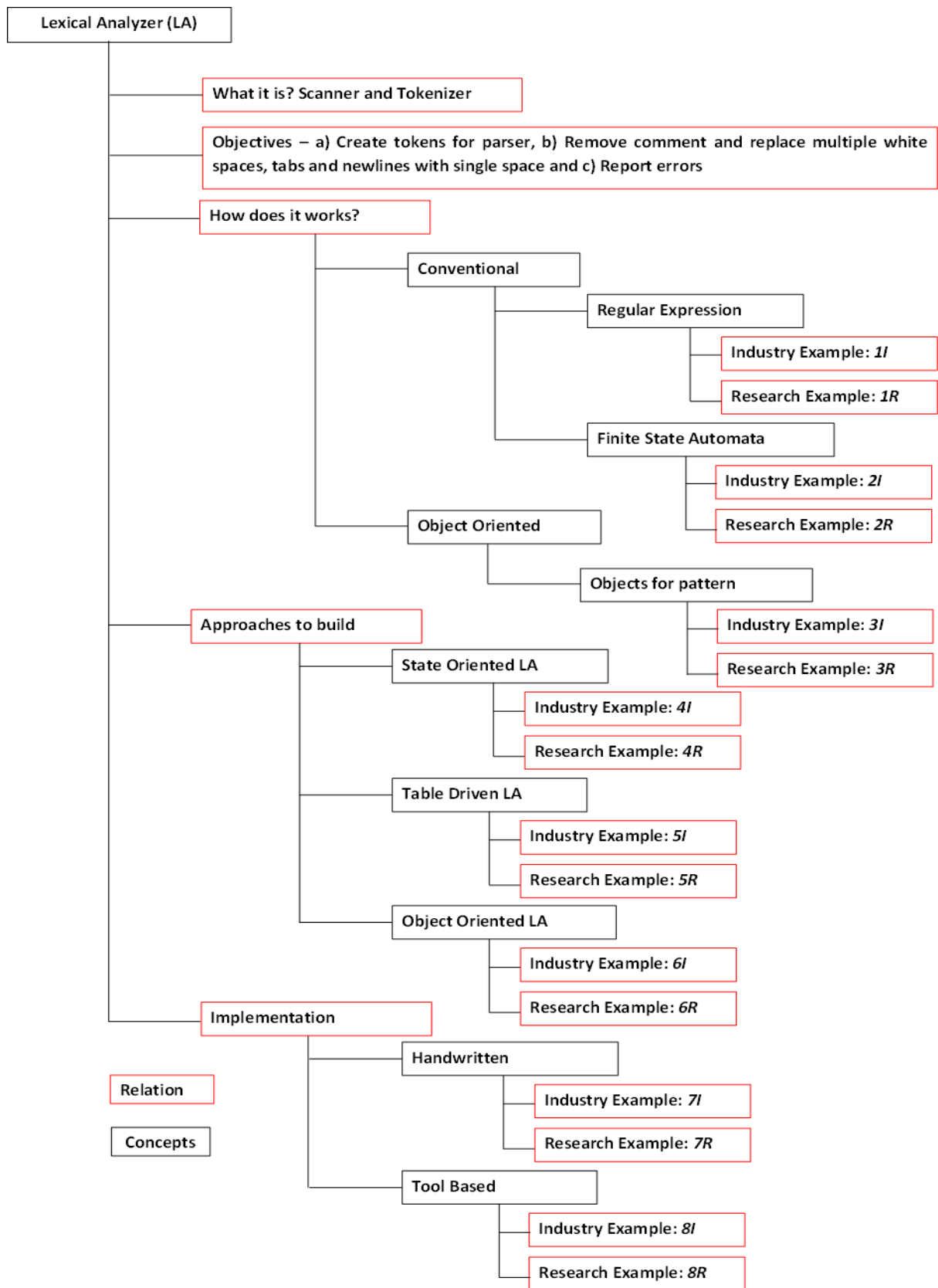


Figure 4. Concepts of Lexical Analyzer

- *2R*: Design a process flow for managing enterprise documents and identify all possible states that a document can be in; and identify the corresponding actions that allow the documents transition between states. Software usually allows auto deleting documents once they have been in a final state for a given number of days, months or years. This is useful for compliance scenarios in which some documents must be kept, archived and later deleted following strict rules following legal requirements [37].
- *3I & 6I*: An object-oriented lexical analyzer for the compiler construction can simplify the design effort, and it permits code re-use [38].
- *3R & 6R*: An industry that handles big data and needs analysis can use the object-based scanner to recognize.
- *4I*: The state oriented lexical analyser can be designed to detect the SQL injection attack in a database [39].
- *4R*: Use the state oriented lexical analyser for sentimental analysis of entertainment industry for prediction of movies. Identify the states and transition for the prediction of the movie [40].
- *5I*: Design the table-driven lexical analyser tool to identify the patterns of a Go language.
- *5R*: Design the table-driven Lexical analyser to analyse the log file of an online product selling Industry.
- *7I*: Design the handwritten scanner using the state oriented/table driven/object-oriented lexical analyser to extract the useful information from an email message to categorize the emails.
- *7R*: Design the handwritten tokenizer using the state-oriented, table-driven or object-oriented lexical analyser to analyse natural language data.
- *8I*: Design the automated tokenization tool for an online product selling Industry that can be reused for the cyber security industry that needs to scan the packets datagram to identify the attack if any.
- *8R*: Design the automated lexical analysis tool for sentiment, social cognition, and social-order analysis [41].

Other phases of compiler. The Figure 5 to 9 respectively shows the extended concept map of Syntax Analyzer, Semantic Analyzer, Intermediate Code Generation, Code Optimization and Code Generation phase. The Industry and Research example for the core concepts need to be discussed in the class when the respective

concept is taught to the students. This will motivate students to study Compiler Design subjects with interest.

5. Evaluation

The proposed concept map, directed concept graph and concept relation weight based teaching methodology was implemented/experimented with while teaching Compiler Design course and evaluated its significance through students' self-designed problems, assignments and feedback.

5.1. Students Feedback

To validate the proposed teaching methodology, the students' opinion about the Compiler design subject was collected at the end of the course through anonymous survey using the same three questions discussed in section 1. The number of students from the Batch - I (2019) who participated in the survey is 60. The outcome is shown in Figure 10, which also includes the survey outcome of the Batch - I (2019) students at the beginning of the course. The difference in the survey outcome indicates that the students' mindset about the Compiler Design subject is changed positively, and that the proposed teaching methodology influenced students. It is very clear from the post-event outcome that students are largely interested in the subject, and they realized that the subject's requirement in research is more, however, they found the subject's requirement in the Industry is less. To overcome this issue, instructors needs to discuss more assignments and problems from the industry domain.

Students feedback reliability justification. In general, students always feel that the course is about to be completed and the grade will be generated so accordingly let us answer the survey questions to gain higher grades. This way of answering will give a better results for the proposed teaching methodology but it will not be fact (authentic) and the whole experiment is useless. But in the survey, the results seems authentic since students' view of placement are reduced at the same time, it is massively increased in interest and research. Hence, the students' feedback is not biased considering grades.

Sign Test. To test the significance of the proposed teaching methodology, the survey outcome is applied to the *sign test*. The survey taken was anonymous so mapping the pre and post-event outcome with a participant is impossible. Hence, random mapping was used to perform the test. Also, the participants in the beginning (pre-event) are 92 and at the end (post-event) are 60 so randomly, 32 participants' feedback are removed from the pre-event survey list. The survey at the beginning (x) and the end (y) had same three

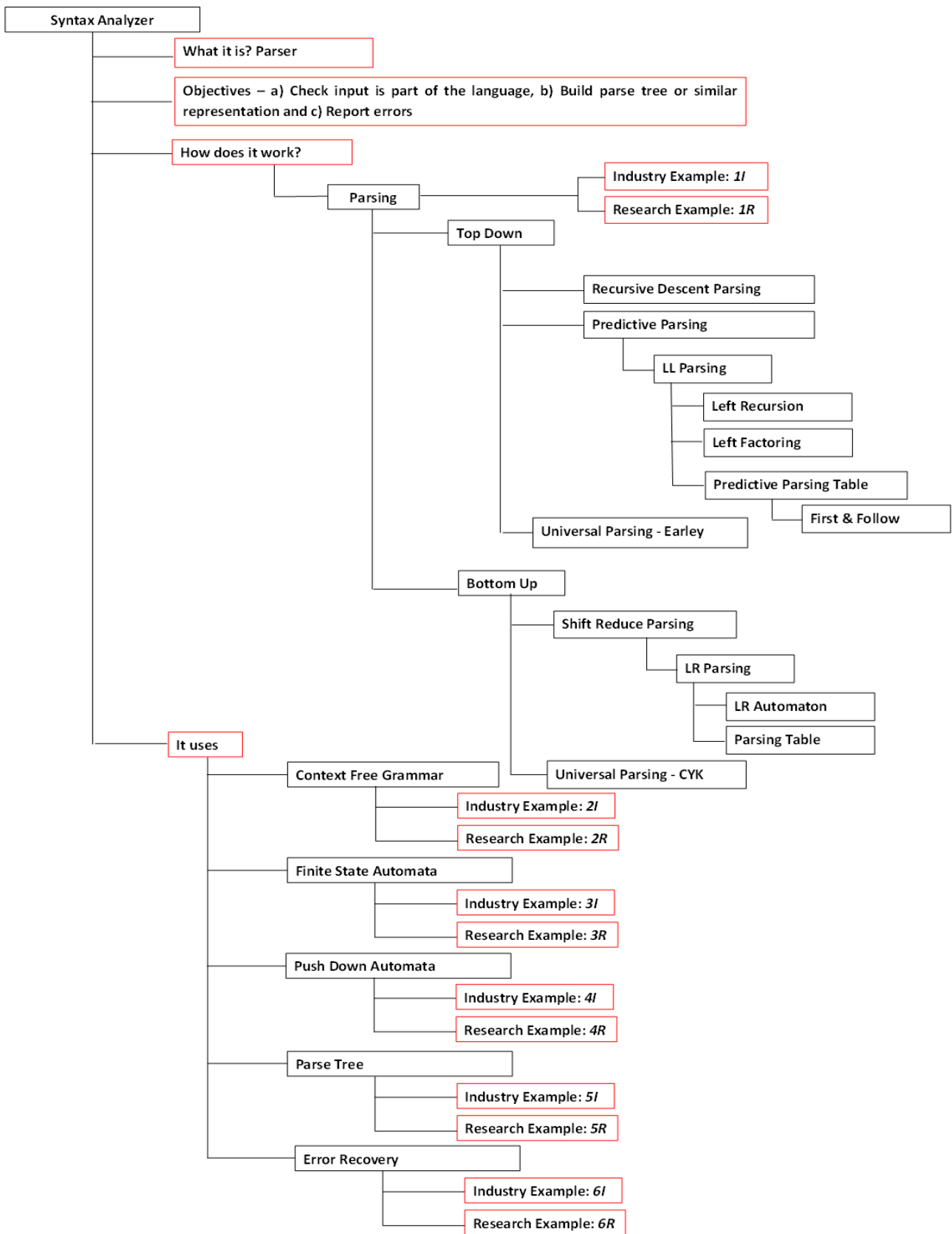


Figure 5. Concepts of Syntax Analyzer

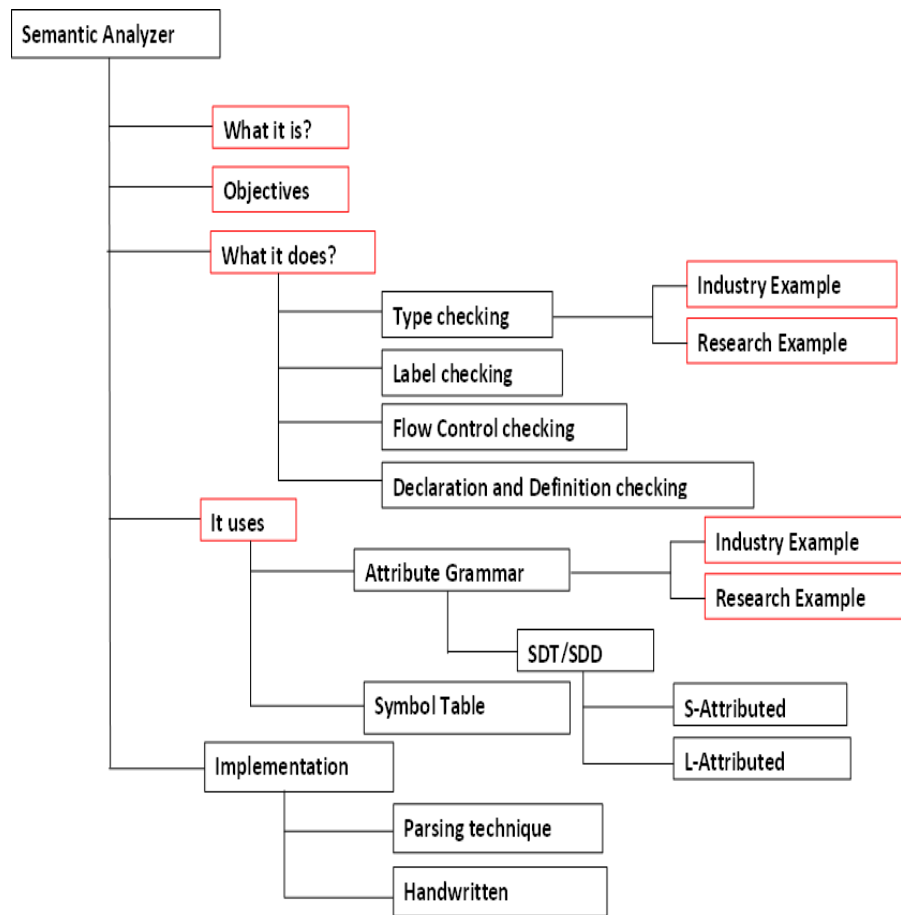


Figure 6. Concepts of Semantic Analyzer

questions and equivalent students' answers. The *sign test* attribute discussed in section 3.2 and its values along with computed cumulative binomial probability (*bp*) and two-tailed test value are given in Table 4.

It can be concluded from the results in Table 4 that the proposed teaching methodology had a positive effect on the students since the two-tailed binomial probability value *bp* is less than significance level 0.05. Hence the null hypothesis H_0 is rejected, and the alternate hypothesis H_a is accepted. The proposed teaching methodology changed the students' negative notion about the subject towards the positive side. That is, students understand the need of the subject concerning to Industry and Research requirements, shown interest and get motivated.

5.2. Students Assignment

As a part of an evaluation and experiment, students were asked to form a five member team and prepare assignments on each core concept of the compiler with respect to both Industry and Research. Marks were awarded while grading to motivate the students to actively participate in the assignment preparation. The

instructors evaluated students' submitted assignments according to relevance with the concept. Table 5 shows the Batch - I (2019) assignment evaluation results in percentage with respect to the relevancy of assignments with the concept. The results shows that for few concepts students' team prepared the relevant assignments, and for other concepts, they are weak. For example, only 45% and 50% of students teams are able to self design the Industry and Research assignments, respectively for the Lexical Analyzer phase concepts. Hence, the Instructor have to identify few more example assignments on these concepts and discuss in the class to improve the students understanding level. These results helps Instructor to identify the level of students and their understanding of concepts accordingly guide them to understand the concept clearly and self design the relevant industry and research assignments.

6. Proof and Limitation

The Lemmas are proposed to prove the impact of the proposed teaching methodology among students. The

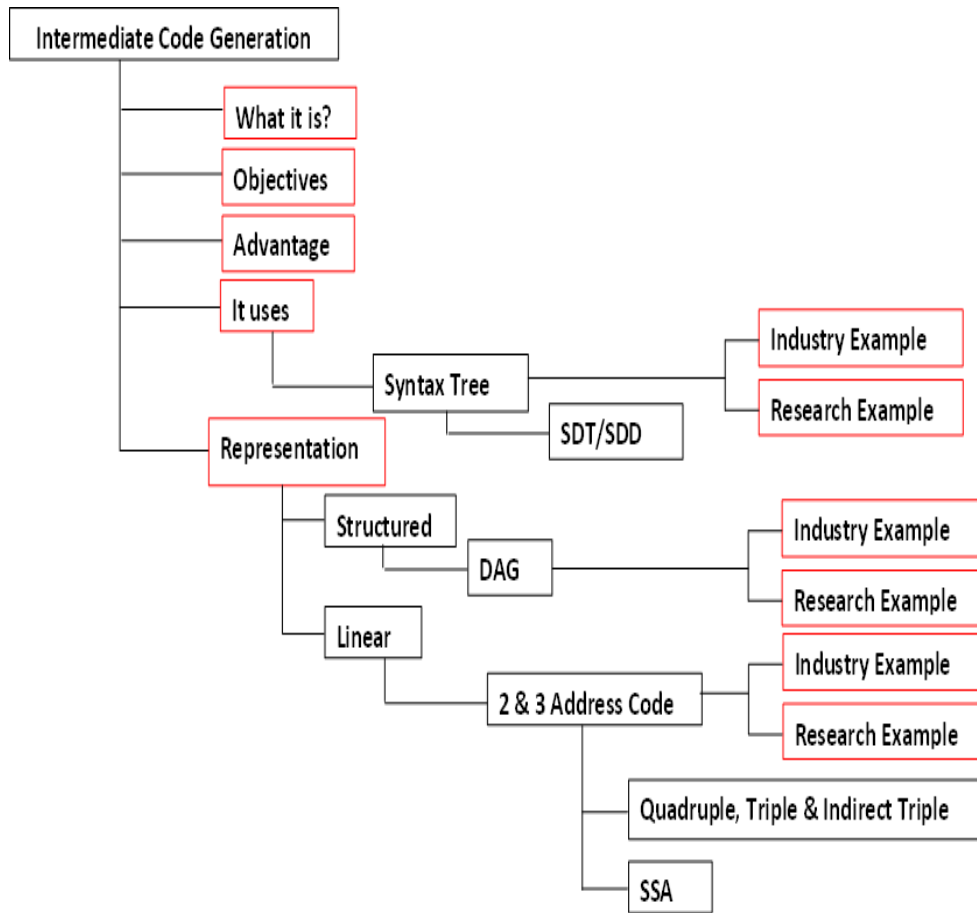


Figure 7. Concepts of Intermediate Code Generation

Table 4. Sign Test Results

Attributes	+	-	0	<i>n</i>	<i>k</i>	<i>bp</i>	Two-tailed
Values	36	16	8	52	16	0.00389	0.00779

Table 5. Compiler Design Assignment Evaluation Results

Concept-Phase	Industry	Research
Lexical Analyser	45%	50%
Syntax Analyser	65%	55%
Semantic Analyser	80%	80%
Intermediate Code Gen.	65%	50%
Code Optimization	70%	65%
Code Generation	65%	65%

notations considered to prove the Lemma are given in Table 6.

In any subject, a class can have N number of students with different categories. The first category of students studies the subject just to pass and get a

degree. The second category of students concentrates only on the subjects which help in industry placement, and consider other subjects are just to complete the degree. The third category of students interested in doing research will concentrate on subjects they feel are required for Research. The fourth category of students focuses on all subjects and studies thoroughly. The fourth category of students are the outliers (\mathcal{P}), who do not need any motivation, but the other three categories need motivation and different teaching methodology. Hence, only two categories (one needs motivation and another does not need motivation) of students considered for the proof. The set of students \mathcal{S} in equation 2 is the two categories of students i.e \mathcal{M} and $\mathcal{P} = \mathcal{N} - \mathcal{M}$ but for further proof \mathcal{S} is considered with one category of students i.e \mathcal{M} . In undergraduate engineering program, there are four years of students

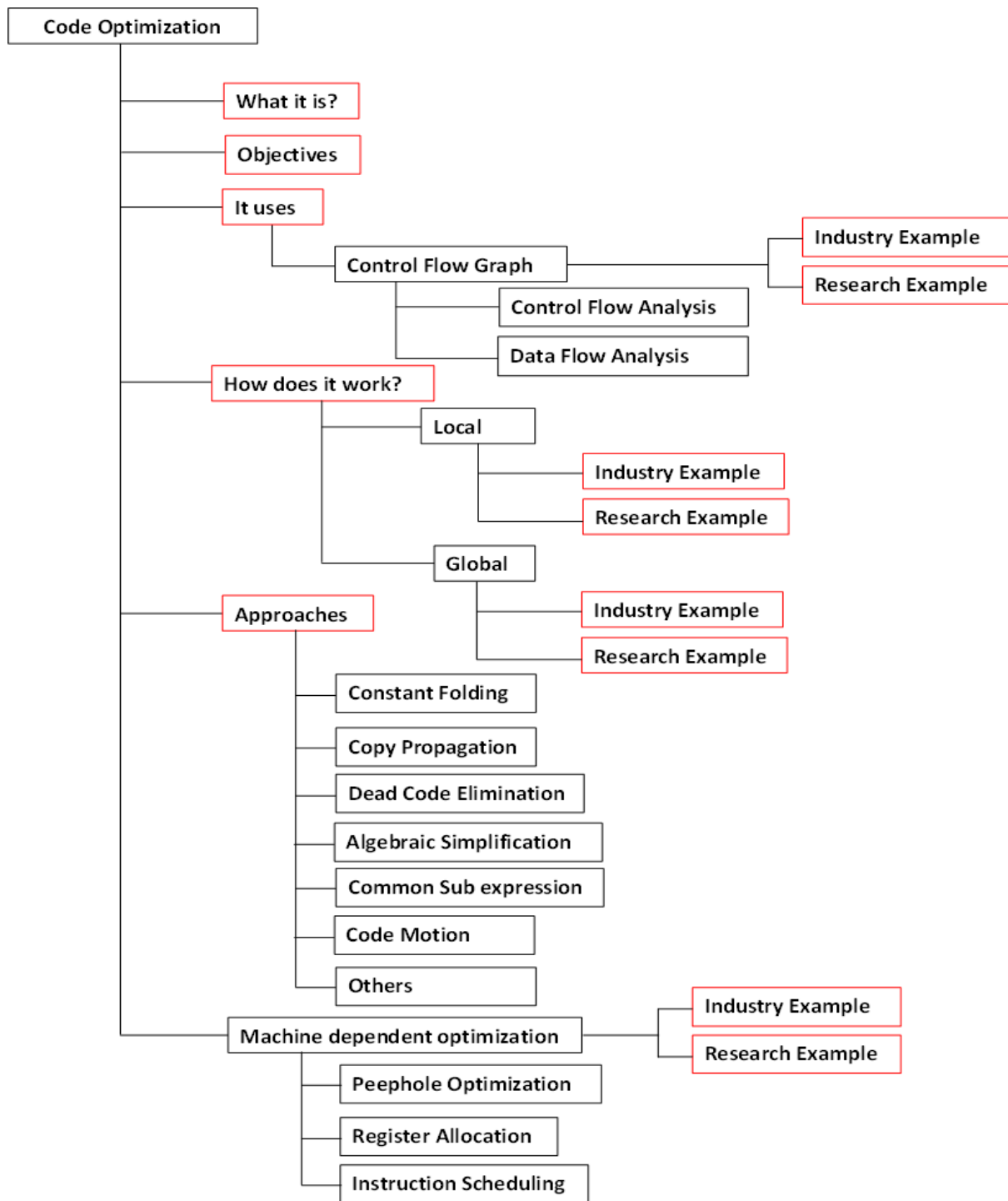


Figure 8. Concepts of Code Optimization

and the junior batch students are influenced by senior batch students as in equation 3.

$$S_b \subset S_1, S_2, S_3, S_4 \text{ where } S_1 \leftarrow S_2 \leftarrow S_3 \leftarrow S_4 \quad (3)$$

Postulate 4. A student is influenced by the senior students and placement record. ($s \leftarrow SS$)

$$S \subset \{s_1, s_2, s_3, \dots, s_M, s_{M+1}, \dots, s_N\} \text{ where } N - M \ll M \quad (2)$$

Postulate 5. A student believes in seniors as well as Instructor but has more belief in senior student than

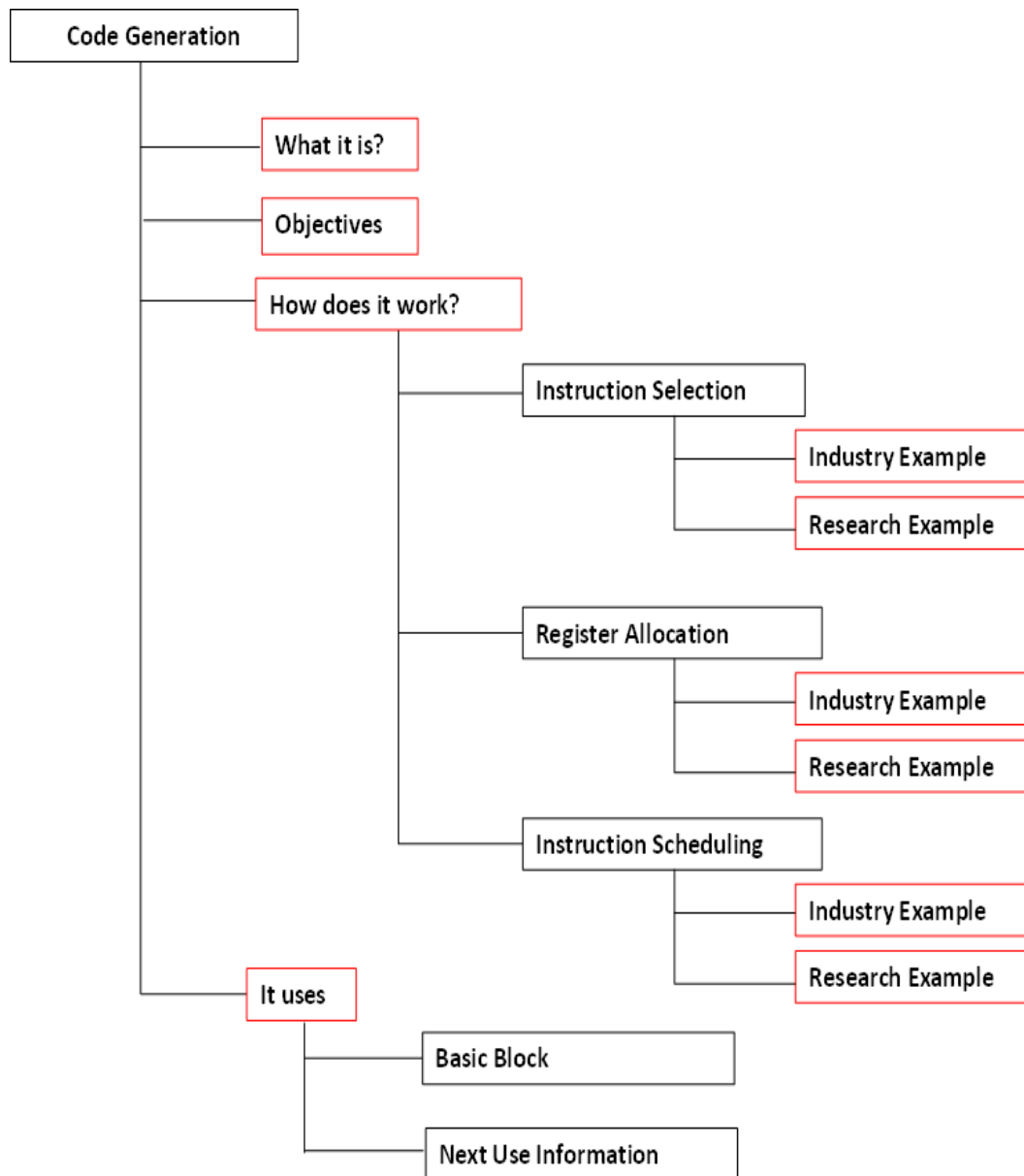


Figure 9. Concepts of Code Generation

Instructor. That is $\mathcal{S}_1 \gg \mathcal{S}_2$ and $\mathcal{S}_1 \gg \mathcal{I}$ but $(\mathcal{S}_1 \gg \mathcal{S}_2) > (\mathcal{S}_1 \gg \mathcal{I})$.

Postulate 6. In most of the cases, student mindset about a subject is fixed before the beginning of the class.

Postulate 7. Linking each concept with present Industry and research requirements can make students know a subject's need.

Lemma 2. If Instructor follows only textbook-based teaching, encouraging a student towards a subject is challenging.

Proof. Given Postulate 4, 5 and 6, encouraging a student with the textbook teaching is a challenging task for

the Instructor if a subject is not directly visible as a requirement for placement and research □

Lemma 3. If Instructor follow the proposed teaching methodology, the student mindset about the subject will be positive and motivated.

Proof. Given postulate 7, students' mindset about a subject will be positive and motivated towards the subject. □

Lemma 4. If the Instructor changes the notion about a subject once, it will have consequences for the junior students.

Table 6. Notations

Notation	Description
S	Students study a subject
SS	Students already studied a subject
S_b	Year wise batch students
\mathcal{I}	Instructor of a subject
\mathcal{N}	Total number of students
\mathcal{M}	Number of generic students who needs motivation
\mathcal{O}	Outliers $\mathcal{N} - \mathcal{M}$, concentrate on subject irrespective of the teaching methodology and seniors comments
$a \leftarrow b$	b influences a
$a \gg b$	a believe b
$x \ll y$	x is very lower than y

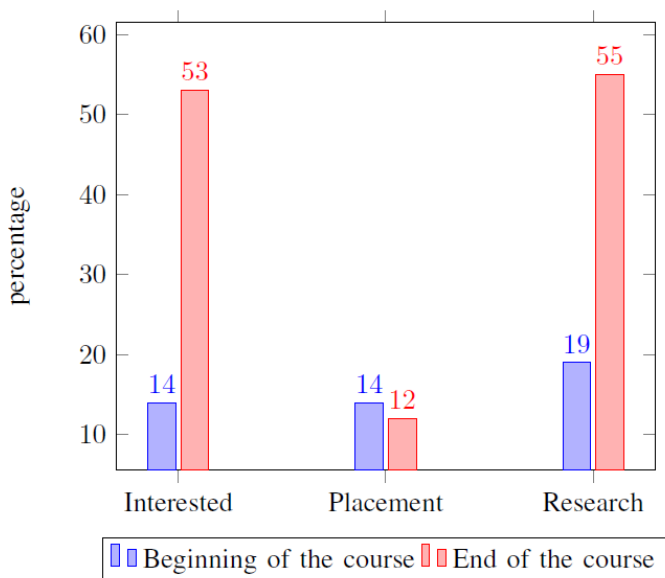


Figure 10. Batch - I (2019) Students opinion (feedback) survey results comparison - Compiler Design

Proof. Given postulate 4 and 5, the junior student notion about a subject can be changed if the senior student is motivated toward the respective subject. \square

The survey results in Figure 10 supports the Lemma 2 and 3. The survey results in Table 2 is not supporting the Lemma 4 however, after two or three batches, Instructor can realize the impact of Lemma 4.

6.1. Limitation

Deciding on the number of teaching hours including lectures, tutorials and practicals hours is more important for a subject since all topics need to be taught to the

students within a semester. The academic committee of an Institute decide credit hours according to the content weight of the subject while creating the curriculum, thus making it possible to complete all topics in a semester. The credit hours vary from a subject to subject and nature of the subjects such as core, elective, add-on, practical oriented, etc. The proposed teaching methodology covers various aspects, including the student assignment and problem evaluation, re-training the students and re-evaluation, and discussing each concept of a subject by following the proposed pyramid model. This teaching methodology needs more time due to multiple learning level (Below Average, Average and Above Average) of students in the class. In some cases the time or total duration of the semester may not be sufficient to complete all the subject topics following the proposed methodology. Hence, the proposed methodology needs to be optimized further to make it effective in the allotted time without affecting the semester’s regular exercises such as exams, result declaration, etc. There is a possibility that the proposed teaching methodology can be applied entirely to the subjects that are not having the open visibility of Industry and Research need for others with a reduction. For example, Machine Learning course has open visibility because a lot many research paper like [42] [43] can be found easily and more work available on it thus we can optimize the proposed teaching methodology. However, the effectiveness should not be affected.

7. Conclusion

This paper analysed various teaching methodologies and identified the three research questions that need to be addressed to improve the quality of teaching and understanding level of students. The proposed teaching methodology using the concept map, concept

order queue, pyramid structure, Industry and Research assignments and problems addresses the questions. The methodology advances the teaching practice compared to the existing techniques. The Compiler Design subject is chosen for the experiment of the proposed teaching methodology since many students pre-decided that this subject is of less scope in the research and development. Teaching Compiler Design through the proposed teaching methodology made students understand the importance of Compiler Design subject in the Industry and research domain, thus motivated them to self-design the problems and assignments and be attentive in the classroom. The *sign test* performed on the students' feedback collected through a survey at the beginning and the end proves the significance and impact of the proposed teaching methodology. This methodology can also be used for other subjects; however the discussed limitation needs to be addressed. The future work of this paper is to optimize the proposed teaching methodology, possibly by doing parallel activities to make it effective in the available time. Also to experiment with other batches and subjects that are considered by the students' as not or less required for their Research or Industry placement or both.

Acknowledgement

Authors would like to thank the students those who are participated in the survey.

References

- [1] VRIES, M.J.D. and STROEKEN, J.H.M. (1996) Developing engineering students' research and technology assessment abilities **6**: 203–219.
- [2] NOVAK, J.D. and NAS, A.J.C., The theory underlying concept maps and how to construct them.
- [3] NOVAK, J.D. (1977), A theory of education.
- [4] C. CHIOUA, L.T. and LEE, L. (2015) Effects on learning of multimedia animation combined with multidimensional concept maps **80**: 211–223.
- [5] A. M. B. PAVANI, G.P.T. and CALLIARI, F. (2020) Concept maps and self regulated learning: an exploratory study. In *2020 IEEE Global Engineering Education Conference (EDUCON)*: 1149–1158.
- [6] LIU, H. (1993) Software engineering practice in an undergraduate compiler course **36**(1): 104–108.
- [7] A. DEMAILLE, R.L. and PERROT, B. (2008) A set of tools to teach compiler construction. In *13th Annual Conference on Innovation and Technology in Computer Science Education (ITICSE'08)*: 68–72.
- [8] XU, L. and MARTIN, E.G. (2006) Chirp on crickets: Teaching compilers using an embedded robot controller. In *37th SIGCSE technical symposium on Computer science education*, **38**: 82–86.
- [9] HENRY, T.R. (2005) Teaching compiler construction using a domain specific language. In *36th SIGCSE technical symposium on Computer science education*, **37**: 7–11.
- [10] E. WHITE, R.S. and STEWART, N. (2005) Hide and show: Using real compiler code for teaching. In *36th SIGCSE technical symposium on Computer science education*, **37**: 12–16.
- [11] VEGDAHL, S.R. (2000) Using visualization tools to teach compiler design. In *Fourteenth Annual Consortium on Small Colleges South eastern Conference (CCSC '00)*: 72–83.
- [12] MERNIK, M. and ZUMER, V. (2003) An educational tool for teaching compiler construction **46**(1): 61–68.
- [13] SHAPIRO, H.D. and MICKUNAS, M.D. (1976) A new approach to teaching a first course in compiler construction. In *ACM SIGCSE-SIGCUE technical symposium on Computer science and education*: 158–166.
- [14] RUCKERT, M. (2007) Teaching compiler construction and language design: Making the case for unusual compiler projects with postscript as the target language. In *ACM SIGCSE-SIGCUE technical symposium on Computer science and education*, **39**: 435–439.
- [15] KUNDRA, D. and SUREKA, A. (2016) Application of case-based teaching and learning in compiler design course. In *arXiv:1611.00271v1 [cs.PL]*.
- [16] KUNDRA, D. and SUREKA, A. (2016) An experience report on teaching compiler design concepts using case-based and project-based learning approaches. In *International Conference on Technology for Education (T4E 2016)*.
- [17] WANG, N. and LI, L. (2019) Teaching reform on compiling principal course. In *The 14th International Conference on Computer Science & Education*: 258–261.
- [18] VELÁSQUEZ, J. (2018) bcc: A suite of tools for introducing compiler construction techniques in the classroom **16**(12): 2941–2946.
- [19] K. ABE, T. TATEOKA, M.S.Y.M.K.K. and WATANABE, T. (2004) An integrated laboratory for processor organization, compiler design, and computer networking **47**(3): 311–320.
- [20] VELÁSQUEZ, J. (2018) Automatic assessment of programming projects in a compiler construction course **16**(12): 2904–2909.
- [21] NOLEN, S.B. and KORETSKY, M.D. (2018) Affordances of virtual and physical laboratory projects for instructional design: Impacts on student engagement **61**(3): 226–233.
- [22] M. FRANK, I.L. and ELATA, D. (2003) Implementing the project-based learning approach in an academic engineering course **13**: 273–288.
- [23] J. ÀNGEL VELÀZQUEZ-ITURBIDE, I.H.L. and PAREDES-VELASCO, M. (2017) Evaluating the effect of program visualization on student motivation **60**(3): 238–245.
- [24] I. ESTÉVEZ-AYRES, C. ALARIO-HOYOS, M.P.S.A.P.R.M.C.G.D.L.H.A.P.G. and DELGADO-KLOOS, C. (2015) A methodology for improving active learning engineering courses with a large number of students and teachers through feedback gathering and iterative refinement **25**: 387–408.
- [25] F. ORTIN, D.Z. and CUEVA, J.M. (2007) Design patterns for teaching type checking in a compiler construction course **50**(3): 273–283.
- [26] S. SANGAL, S. KATARIA, T.T.N.G.Y.K.S.A. and CHAKRABORTY, P. (2018) Pavn: a tool to visualize and teach parsing algorithms, education and information technologies **23**(6): 2737–2764.

- [27] RESLER, R.D. and DEEVER, D.M. (1998) Vcoco: a visualisation tool for teaching compilers. In *6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education*, **30**: 99–102.
- [28] KARKARE, A. and AGRAWAL, N. (2016) Parseit: A tool for teaching parsing techniques. In *47th ACM Technical Symposium on Computing Science Education*: 590.
- [29] ERDIL, D.C. (2020) A project-based learning approach to teaching computer architecture **6**(19).
- [30] CONOVER, W.J. (1999) The sign test : 157–176.
- [31] WORM, G. The sign test .
- [32] SUBRAMANIAN, V. and NATARAJAN, K. (2019) Simplification of compiler design course teaching using concept maps. In *7th International Conference on Information Technology in Education (ICITE 2019)*, **9**: 91–102.
- [33] A. V. AHO, M. S. LAM, R.S. and ULLMAN, J.D. (2006) Compilers: Principles, techniques and tools. In *37th SIGCSE technical symposium on Computer science education*.
- [34] Part i: Lexical analysis slides .
- [35] S. BISWAS, D.SENGUPTA, R.B. and HANDIQUE, M. (2016) Text manipulation using regular expressions. In *6th International Conference on Advanced Computing*: 62–67.
- [36] JAMHOUR, A. and GARCIA, C. (2012) Automation of industrial serial processes based on finite state machines. In *20th International Congress of Chemical and Process Engineering*: 186–196.
- [37] RAMIREZ, J. Finite-state machines: Better than flowcharts .
- [38] SCHREINER, A. and KÜHL, B. Object-oriented compiler construction .
- [39] NTAGWABIRA, L. and KANG, S.L. (2010) Use of query tokenization to detect and prevent sql injection attacks. In *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*: 438–440.
- [40] RAJ, M.P.M. and ADITYA, S. (2017) Predictive model for movie's success and sentiment analysis **6**: 1–7.
- [41] S. A. CROSSLEY, K.K. and McNAMARA, D.S. (2017) Sentiment analysis and social cognition engine (seance):an automatic tool for sentiment, social cognition,and social-order analysis **6**: 803–821.
- [42] Z. REN, Y.Z. and WANG, S. (2017) A hybrid framework for lung cancer classification **11**(10).
- [43] S. WANG, P. PHILLIPS, Y.S.B.L.M.Y.H.C. (2018) Classification of alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling **42**(5).