INSTITUTE VISION

To emerge as one of the finest technical institutions of higher learning, to develop engineering professionals who are technically competent, ethical and environment friendly for betterment of the society.

INSTITUTE MISSION

Accomplish stimulating learning environment through high quality academic instruction, innovation and industry-institute interface.

DEPARTMENT VISION

To develop technical professionals acquainted with recent trends and technologies of computer science to serve as valuable resource for the nation/society.

DEPARTMENT MISSION

Facilitating and exposing the students to various learning opportunities through dedicated academic teaching, guidance and monitoring.

PROGRAM EDUCATIONAL OBJECTIVES

1. Lead a successful career by designing, analyzing and solving various problems in the field of Computer Science & Engineering.
2. Pursue higher studies for enduring edification.
3. Exhibit professional and team building attitude along with effective communication.

4.Identify and provide solutions for sustainable environmental development.

Subject Name– Code - Course Outcomes (COs) w.r.t this PBL	
CO #	CO DEFINED
	ASK YOUR FACULTY ABOUT THIS

Project to Program Outcomes (PO) Mapping

Project Name: title (ASK YOUR FACULTY ABOUT THIS)

COURSE	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Subject Name												
Subject Name												

Program outcomes (POs):	
PO1	**Engineering knowledge:** Apply the knowledge of Mathematics, Science, Engineering fundamentals and an engineering specialization to the solution of complex engineering problems
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyse complex Engineering problems reaching substantiated conclusions using first principles of mathematics, Natural sciences and engineering sciences
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the Information to provide valid conclusions
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for Sustainable development
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering Community and with society at large, such as, being able to comprehend and write effective reports And design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and Leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project to Program Specific Outcomes (PSO) Mapping

Program Specific Outcomes (PSOs):	
PSO1	Analyze the problem and identify computing requirements appropriate to its solution.
PSO2	Apply design and development principles in the construction of software systems of varying complexity.

Project Name: title

COURSE	PSO1	PSO2
Subject Name		
Subject Name		

Use the Tick symbol (√) for mapping

Abstract:

The CLR parser stands for canonical LR parser. It is a more powerful LR parser. It makes use of lookahead symbols. This method uses a large set of items called LR(1) items. The main difference between LR(0) and LR(1) items is that, in LR(1) items, it is possible to carry more information in a state, which will rule out useless reduction states. This extra information is incorporated into the state by the lookahead symbol. The general syntax become[A->∝.B , a] where A->∝.B is the production and a is a terminal or right end marker $LR(1) items=LR(0) items + look ahead.

Problem Statement: Designing of Canonical LR Parser

Introduction:

A canonical LR parser or LR(1) parser is an LR(k) parser for k=1, i.e. with a single lookahead terminal. The special attribute of this parser is that any LR(k) grammar with k>1 can be transformed into an LR(1) grammar.[1] However, back-substitutions are required to reduce k and as back-substitutions increase, the grammar can quickly become large, repetitive and hard to understand. LR(k) can handle all deterministic context-free languages.[1] In the past this LR(k) parser has been avoided because of its huge memory requirements in favor of less powerful alternatives such as the LALR and the LL(1) parser. Recently, however, a "minimal LR(1) parser" whose space requirements are close to LALR parsers[citation needed], is being offered by several parser generators.

Motivation:

The following reasons are the motivations to design the Canonical LR parser

- LR parsers can handle a large class of context-free grammars.
- The LR parsing method is a most general non-back tracking shift-reduce parsing method.
- An LR parser can detect the syntax errors as soon as they can occur.
- LR grammars can describe more languages than LL grammars.

Proposed System and Methodology:

We are implementing the Canonical LR Parser. The steps to design CLR is given below.

- Write CFG for the given input symbol
- Check if the grammar is ambiguous or not.
- Add an augmented grammar.
- Create a canonical collection of LR(1) items.
- Draw DFA
- Construct a CLR(1) parsing table.

We are using concepts of System Software and Compiler Design and Web Technologies and its Applications to develop our project 'Design of Canonical LR Parser' .

Construction of the canonical LR parsing table

■Input: An augmented grammar G'.

■ Output: The canonical LR parsing table functions action and goto for G'

■ Method:

1. Construct C= {10, 11..... In}, the collection of sets of LR (1) items for G'.

2. State I of the parser is constructed from li. The parsing actions for state I are determined as follows:

a) If [A-> α. a ß, b] is in li, and goto (li, a) = Ij, then set action [i, a] to "shift j." Here, a is required to be a terminal.

b) If [A-> α., a] is in li, A ≠ S', then set action [i, a] to "reduce A-> α."

c) If [S'-> S., $] is in li, then set action [i, $] to "accept."

If a conflict results from above rules, the grammar is said not to be LR (1), and the algorithm is said to be fail.

3. The goto transition for state i are determined as follows: If goto(li, A)= Ij,then goto[i,A]=j.

4. All entries not defined by rules (2) and (3) are made "error."

5. The initial state of the parser is the one constructed from the set containing item [S'→.S, $].

System Requirement Specifications:

Functional Requirements:

- Input Buffer − It includes the string to be parsed, followed by $, a symbol used as the right end marker to denote the end of the string.
- Stack − It is used to store grammar symbols and states.

Non-Functional Requirements:

- Performance
- Reliability
- Availability
- Security
- Maintainability

Subject Mapping:

System Software and Compiler Design (SSCD):

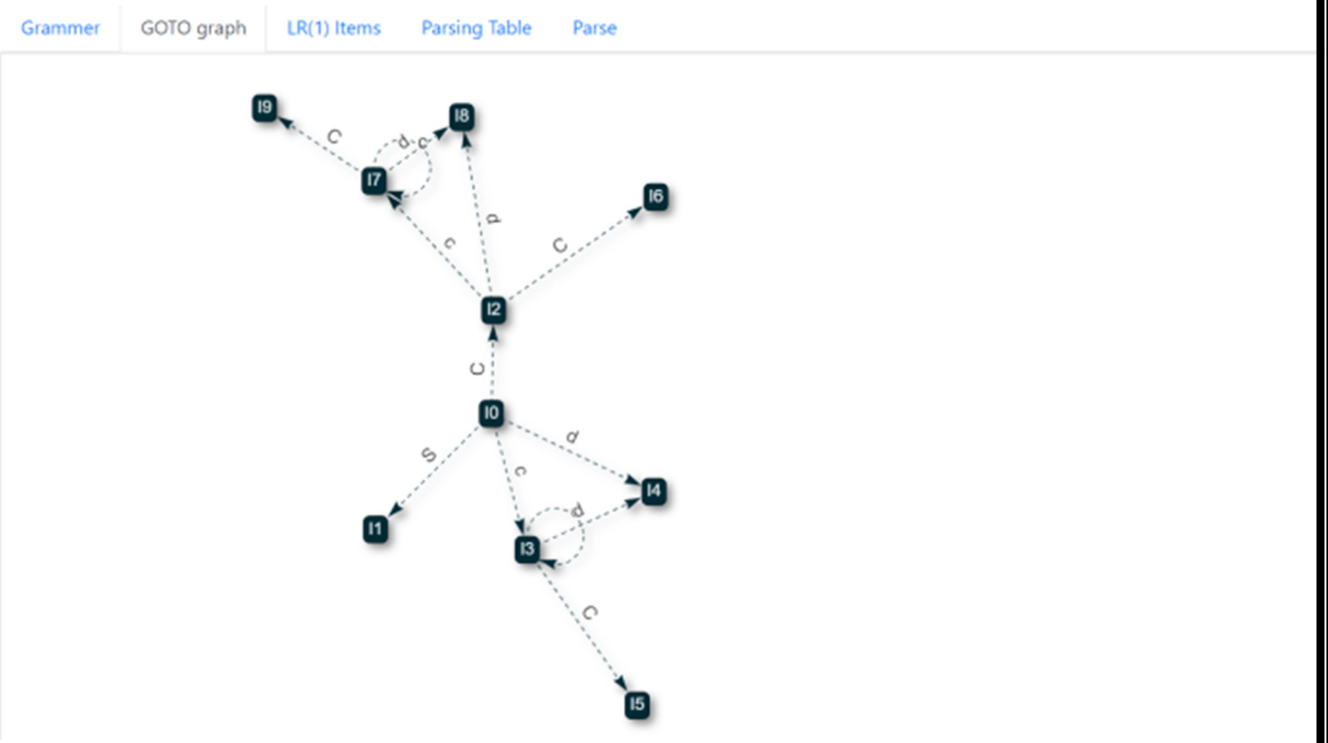
- First and Follow concepts
- LR(1) items
- Canonical LR Parser (LR(1))

Web Technologies and its Applications (WTA):

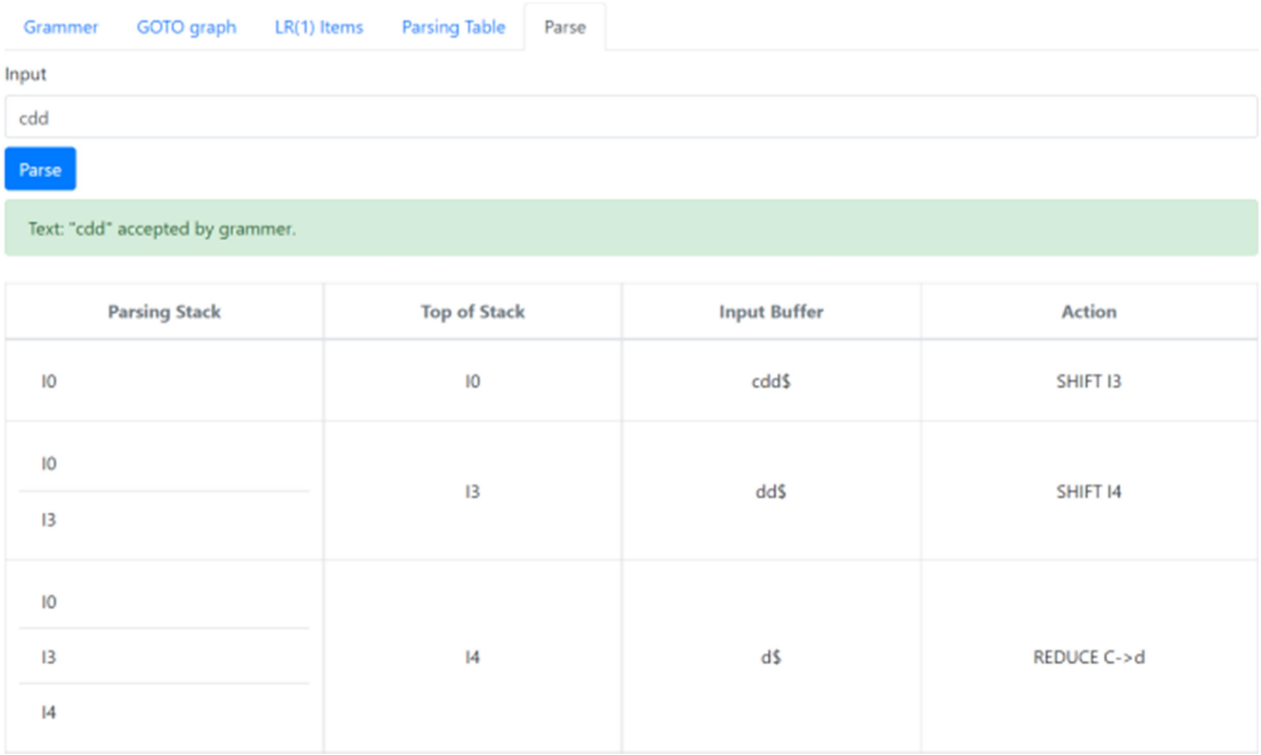
- CSS
- JavaScript
- TypeScript
- HTML

IMPLEMENTATION:

GOTO GRAPH:



OUTPUT:



Parsing Stack	Top of Stack	Input Buffer	Action
I0	I0	cdd$	SHIFT I3
I0 I3	I3	dd$	SHIFT I4
I0 I3 I4	I4	d$	REDUCE C->d

Grammer | GOTO graph | LR(1) Items | Parsing Table | Parse

STATES	ACTION			GOTO		
	c	d	$	S	C	S_
I0	SHIFT I3	SHIFT I4	-	I1	I2	-
I1	-	-	ACCEPT	-	-	-
I2	SHIFT I7	SHIFT I8	-	-	I6	-
I3	SHIFT I3	SHIFT I4	-	-	I5	-
I4	REDUCE C->d	REDUCE C->d	-	-	-	-
I5	REDUCE C->cC	REDUCE C->cC	-	-	-	-
I6	-	-	REDUCE S->CC	-	-	-
I7	SHIFT I7	SHIFT I8	-	-	I9	-
I8	-	-	REDUCE C->d	-	-	-
I9	-	-	REDUCE C->cC	-	-	-

REFERENCES

- https://youtube.com
- https://google.com
- https://geeksforgeeks.com
- https://github.com