**LL(1) Parser**

A Mini Project Report Submitted by

Rahul D Shetty 4NM16CS111

Saurabh D Rao 4NM16CS132

UNDER THE GUIDANCE OF

Mrs. ANISHA P RODRIGUE

Assistant Professor Gd. II

Department of Computer Science and Engineering

in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering in Computer Science & Engineering

from

Visvesvaraya Technological University, Belgaum



 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

N.M.A.M. INSTITUTE OF

TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO

9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

**April 2019**



(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

: 08258 - 281039 – 281263, Fax: 08258 – 281265

**Department of Computer Science and Engineering**

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

CERTIFICATE

**LL(1) Parser**

is bona fide work carried out by

Rahul D Shetty 4NM16CS111

Saurabh D Rao 4NM16CS132



in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

prescribed by Visvesvaraya Technological University,

Belgaum during the year 2018-2019.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD

# 

# **ACKNOWLEDGEMENT**

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We express our deep sense of gratitude and indebtedness to our guide **Mrs. ANISHA P RODRIGUE**, Assistant Professor Gd II, Department of Computer Science and Engineering, for his inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We sincerely thank **Dr. K.R. Udaya Kumar Reddy**, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

We also thank all those who have supported us throughout the entire duration of our project.

Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

Rahul D Shetty 4NM16CS111

Saurabh D Rao 4NM16CS132

# 

# **ABSTRACT**

The purpose of this project is to design lexical analyser and syntax analyser for a LL(1) Grammar. The two stages are the integral part of Analysis phase of a compilation process which involves identifying the tokens of the given program and using these tokens to identify if each of them are syntactically proper based on given production rules. The main program takes in two inputs namely the source program which we need to process and the grammar rules to parse the program. The objective of the project is to generate the parsed sequence which can be further given for the later stages of the compiler.

The grammar that is defined for parsing, should be LL(1) that is to say it should not contain any left recursion and it should be left factored. By using the LL(1) productions, we generate the parse table which has entries for each terminals and non-terminals identified in them. Before the generation of parse table, we identified the FIRST and FOLLOW’s of each terminals using a recursive method. The final stage is the parsing which is done by using the standard LL(1) parsing steps. If the given source code contains some syntax errors, the appropriate line number would be shown. The error handling part of the parser is implemented using Panic Mode recovery.

The outcome of the project is to identify the parsing actions taken by the grammar for proper and invalid source code.

**Table of Contents**

1. Introduction

Compiler

Phases in Compiler

Lexical Analysis

Syntax Analysis

1. Implementation

Token Rules

Lexical Analyser

Production Rules

Syntax Analyser

1. Results
2. Conclusion

The Mini project report should have following components

* Abstract
* Table of contents
* Introduction
* Implementation
* Results
* Conclusion

The mini project report formatting should follow the below rules.

* Font should be “times new roman”.
* Heading fond size is 16pts in capital letter
* Sub heading font size is 14pts.
* Normal text font size is 12 pts

One sample text and its format is provided in next page

**CHAPTER 1**

## **INTRODUCTION**

**1.1 OVERVIEW**

Cheating in Visual Cryptography (VC) is well studied and understood through secret-sharing schemes [1], [2]. VC is a variant of secret sharing. Most cheating attacks in VC are known to be plaintext attacks where the cheaters know the secret image and are able to infer the blocks of victim’s transparency based on the base matrices. It is noticed that cheating is possible in (k, n) and (n, n) VC where k is smaller than n.

There are mainly two types of cheaters in VC.

1. A malicious participant (MP) who is also a legitimate participant, namely MP € P (Qualified participant)
2. A malicious outsider (MO), where MP € P.

A cheating process against a VCS consists of the following 2 phases:

1)**Fake share construction phase**: The cheater generates the fake shares.

2)**Image reconstruction phase**: The fake image seems to appear on the stacking of genuine shares

and fake shares.

In order to cheat successfully, honest participants who introduce their shares for recovering the secret image should not be able to distinguish fake shares from genuine shares. A reconstructed image is perfectly black if and only if the sub pixels associated to a black pixel of the secret image are all black. Most of the VC schemes have the property of perfect blackness.

Some of most common ways how MO and MP cheat visual cryptography are:

1)Cheating a VC by an MP

2)Cheating a VC by an MO

3)Cheating an Extended VCS by an MP.

**Cheating a VC by an MP**

A qualified standard participant called as MP (Malicious Participant) can also be a cheater, where the participant creates a fake share image by using his original share images. By doing so, he will try to cheat the other participants who are genuine because the fake share images generated will be indistinguishable from the original share images and also the decoded output image will be different from the original secret image.

**Cheating a VC by an MO**

A disqualified participant called as MO (Malicious Outsider) will create fake shares by using some random images as the input and will try to decode the original image. The MO will try to create fake shares of different sizes because the size of the original share can vary.