목차

[서론 1](#_Toc119580664)

[1) 과제 소개 1](#_Toc119580665)

[2) 구현된 부분과 구현되지 않은 부분 1](#_Toc119580666)

[문제 분석 1](#_Toc119580667)

[1) 구현 대상 1](#_Toc119580668)

[2) Grammar rule분석 2](#_Toc119580669)

[설계 2](#_Toc119580670)

[1) 주요 자료구조(syntax tree등) 2](#_Toc119580671)

[2) 프로그램 module hierarchy 및 module에 대한 설명 2](#_Toc119580672)

[수행 결과 2](#_Toc119580673)

# 서론

## 1) 과제 소개

지난 1차 과제는 c언어를 사용해 어휘 분석기를 구현. 이번 2차 과제는 flex를 사용해 어휘 분석기를 구현한 후 c언어를 사용해 구문 분석기를 구현하는 것이다.

## 2) 구현된 부분과 구현되지 않은 부분

구현되지 않은 부분 : 우선 lex를 사용해 token을 만들었을 때 token이 잘못된 입력인 경우를 배제하였다.

# 문제 분석

## 1) 구현 대상

재귀적 하강 구문 분석은 재귀적 프로시저의 집합으로 구성된다. 각 논터미널 기호에 대한 프로시저를 작성하고 터미널 기호에 대한 프로시저를 작성한 다음 이를 통합함으로써 구현이가 가능하다. 터미널 기호에 대한 프로시저는 생성 규치에 있는 터미널 기호와 입력 기호가 같은지 비교하여 같으면 다음 입력 기호를 읽게 하면 된다.

구현해야 할 기능은 크게 3가지이다. 1) 연산, 2) abstract syntax tree 3) symbol table.

Abstract syntax tree를 구성할 경우 연산과 symbol table은 따라오는 것이기에 abstract syntax tree만드는 것을 목표로 하였다.

## 2) Grammar rule분석

# 설계

## 1) 주요 자료구조(syntax tree등)

## 2) 프로그램 module hierarchy 및 module에 대한 설명

# 수행 결과