**프로그래밍언어 Project2 Lexer 보고서**

2017313213 박경태

1. 과제 개요

본 과제는 주어진 언어를 컴파일하는 컴파일러의Lexer, Parser, Code Generator 를 설계하는 팀프로젝트이다. 컴파일러는 아래 사진과 같은 구조를 가지고 있으며, 아래의 사진단계로 설명하고자 한다.

Diagram

Description automatically generated

2. Lexical Analyzer 설명

Diagram

Description automatically generated

Lexical Analyzer는 주어진 소스프로그램을 토큰으로 분리하여 parser에 전달해주는 역할을 담당한다.

Code Stream 을 Token Stream 으로 바꿔야 하는데, 이 과정에서 유의할 점이 더러 있다.

예를 들어, 기존 C++ 문법같은 경우 pair<int,int>> 에서 >> 라는 닫힘 괄호를 "parenthesis"가 아닌 "operator" 역할로 오해하여 parser에 전달시키면 안될 것이다. 이러한 오해는 Token stream 으로 변환하는 과정에서 해소되게 된다. 그리고 이러한 작업은 lexer 의 역할이다.

주어진 문법은 상대적으로 매우 간단한 편이라, 그런 오해가 벌어질 염려는 적다.

Text

Description automatically generated

lexical.py 에서는 input된 code를 token 으로 변환하는 작업을 구현한다.

token 은 리스트로 구현되어있으며, 리스트에는 길이 2의 리스트가 들어가게 된다.

최종적으로 분류하는 가능한 lexeme 속성의 종류는 아래와 같다.

number: 숫자를 의미

type: 자료형을 의미

statement: if, then, else 등의 구문을 의미

EXIT: 종료구문을 의미

word: 변수명을 의미

bracket: 열고닫는괄호

operator: 연산자

comma or semicolon: 괄호 및 세미콜론

우리가 a=110; 라는 문법을 받았는데, a=1 로 끝나면 안될 것이다.

따라서, 끝까지 읽어주는 read\_full\_digit 함수를 만든다.

Graphical user interface, text, application

Description automatically generated

마찬가지로, 우리가 str = alpha 을 입력받았는데 str=a 만 입력되면 안될 것이다.

따라서, read\_full\_string 함수도 만든다. Graphical user interface, text, application

Description automatically generated

==와 같이 두 글자 이상의 길이를 갖는 operator 는 없으니(있다면 =와 ==를 구별하여 식별할 수 있어야 하므로) 따로 operator 를 분류할 필요는 없어서 구현 상 편리함이 있다.

lexer 역할을 담당하는 lexical 부분은 아래와 같다.

Text

Description automatically generated

숫자가 들어오면 반드시 숫자이므로 (변수는 숫자로 시작할 수 없다) number 로 받아들여준다.

알파벳으로 시작하면 여러 경우의 수가 가능하다.

type 또는 statement 또는 EXIT 또는 word 의 경우가 가능한데, 각각 읽힌 token 이 어느 것이냐에 따라 구분이 가능하다.

공백 또는 \n, \t 의 경우 무시한다. whitespace 는 문법에 무의미하다.

만일 숫자도 아니고 알파벳도 아니라면 연산자 또는 괄호나 특수문자일 것이다.

각각 케이스를 분류해준다.

그리고, tokens 배열에 append 해준 뒤 함수를 종료한다.

­이로써 tokens 에는 symbol table 이 저장되게 되며, 이 symbol table 은 parser 로 넘어가게 된다.

테스트한 코드는 아래와 같으며,코드에서 추출된 토큰들은 아래에 첨부한 바와 같다.

Text

Description automatically generated

Text

Description automatically generated with low confidence