
Project2

Lexer & Parser

- Lexer

- C 소스 코드를 읽어 parser 에게 적절한 정보를 넘길 수 있는 lexer 구현
 - Token을 리턴하도록 변경
 - Lexeme 값 (yyval)은 이 프로젝트에서 고려하지 않음
- 이전 프로젝트에 이어서 구현해도 되고, 새로 주어진 스켈레톤을 써도 됨

- Parser

- Bison declarations section
 - token 및 token(terminal) 들의 precedence & associativity 정의하기
- Grammar rules section
 - C 의 grammar 에 맞는 rule 작성하기
 - Reduce 발생 시 사용된 production 을 stdout 에 출력하도록 action 정의

Output Format

- printf 사용하여 test 폴더 내의 예시와 동일하게 출력
- 테스트 방법 예시) ./subc test.c > result

```
int main(){
    int a;
    char b;

    a = 10;
    b = 5;

    if (a==10 || b==5) {
        return 1;
    } else {
        return 0;
    }
}
```

Input



```
$ ./subc ../examples/ex_format/test.c
ext_def_list->epsilon
type_specifier->TYPE
pointers->epsilon
func_decl->type_specifier pointers ID '(' ')'
def_list->epsilon
type_specifier->TYPE
pointers->epsilon
def->type_specifier pointers ID ';'
def_list->def_list def
type_specifier->TYPE
pointers->epsilon
def->type_specifier pointers ID ';'
def_list->def_list def
stmt_list->epsilon
unary->ID
unary->INTEGER_CONST
binary->unary
expr->binary
```

Output

Tips

- Lex 소스 코드 (.l file) 작성 시

- '|' 활용하여 묶어 처리할 수 있는 token 은 regular expression 하나로 작성
 - EX) "<"|"<="|">"|">=" { return RELOP; }
- Name 없는 token 처리
 - Lex에서는 yytext[0] 을 리턴
 - Yacc에서는 '+'와 같은 형태로 Grammar Rule 에서 token 입력 인식

- Parser 소스 코드 (.y file) 작성 시

- Unary operation 관련 conflict 해결하기 위해 **%prec** 활용
- IF-ELSE 관련 %precedence 로 선언해도 충분

Tips

- 주어진 C grammar rule 은 conflict 가 발생하도록 설계됨
 - 발생한 conflict 를 해결하기 위해 **Precedence & Associativity** 활용
 - **Grammar 의 production 을 변경하는 것은 금지**
- Hint
 - bison으로 subc.y 파일을 컴파일하면 subc.output 파일이 생성됨
 - bison -**v**d subc.y
 - subc.output 파일을 열면 conflict에 대한 정보를 얻을 수 있음
 - 혹은 bison 의 -Wcounterexamples 옵션 활용
 - bison -**Wcounterexamples** -d subc.y

Submission

- 제출기한

- 5/9 (금요일) 23:59

- 하루 delay 마다 20% 씩 감점

- 제출방법

- lms 의 레포트 Project2 통해 제출

- 제출파일

- 'src' directory 안의 작성한 파일들을 zip 으로 압축하여 제출

- Zip 파일 이름: project2_학번.zip

- Ex) project2_202012345.zip

Notice

- **Project2 document (pdf)** 기준으로 구현 및 채점
- Ubuntu 환경에서 bison install 필요
 - `$ sudo apt install bison`
- **채점 기준 (채점용 c input file)**
 - C의 주석(comment)은 포함하지 않음
 - `..'` operator 도 포함하지 않음
 - 문법 오류가 없는 c code
 - test 코드보다 더 복잡한 c code로 테스트 예정
 - Project2 document (pdf) 에 명시된 Grammar 및 token 만 사용