Compiler Assignment #2

(LR Parser)

소프트웨어학부

컴파일러

유재우 교수님

20201758

이창준

2025.03.16

1. 개발 환경

CPU : Apple M1 APL1102 SOC

운영체제 : macOS 14.5

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

2. 문제해결 방법

1) TT

float과 int를 담아두기 위한 구조체이다. isint를 통해 union이 float인지 int인지 구분할 수 있다.

2) int\_stk

int형을 담아두기 위한 stack이다..

push, pop, top 함수를 통해 동작하며, 각각 값 넣기, 값 빼기, 맨 위에 값 가져오는 기능을 수행한다..

3) TT\_stk

TT형을 담아두기 위한 stack이다.

push, pop, top 함수를 통해 동작하며, 각각 값 넣기, 값 빼기, 맨 위에 값 가져오는 기능을 수행한다..

4) TT\_plus(TT, TT)

TT형의 덧셈 연산 위한 함수이다.

1번 파라미터와 2번 파라미터에 대해 4가지 경우의 수를 구분하여 연산한 후, 연산된 값을 리턴한다.

5) TT\_star(TT, TT)

TT형의 곱셈 연산 위한 함수입니다.

1번 파라미터와 2번 파라미터에 대해 4가지 경우의 수를 구분하여 연산한 후, 연산된 값을 리턴한다.

6) yylex isdigit인 경우

처음 isdigit인 경우, isdigit을 만족하지 않는 ch를 찾을 때 까지 값을 입력받는다.

이후 마지막 ch가 실수인지 판별하기 위해 .을 검사한다..

.이라면 마찬가지로 isdigit을 만족하지 않는 ch를 찾을 때 까지 값을 입력받는다. 이때, 이 과정 속에서 . 을 또다시 만난다면 에러를 출력한다.

3. 결과 분석

1) 1 + 2

correct answer : 3

answer : 3

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

2) 3 \* 4

correct answer : 12

answer : 12

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

3) 5 + 6 \* 7  
correct answer : 47

answer : 47

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

4) 8 \* 9 + 10

correct answer : 82

answer : 82

텍스트, 폰트, 화이트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

5) 1.5 + 2.5  
correct answer : 4.0

answer : 4.000000

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

6) 3.0 \* 4.2

correct answer : 12.6

answer : 12.599999

텍스트, 폰트, 화이트, 그래픽이(가) 표시된 사진

자동 생성된 설명

7) 5.5 + 6.5 \* 7.7

correct answer : 55.55

answer : 55.549999

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

8) 8.8 \* 9.9 + 10.1

correct answer : 97.22

answer : 97.2199994

텍스트, 폰트, 화이트, 그래픽이(가) 표시된 사진

자동 생성된 설명

9) 1 +2 + 3 + 4

correct answer : 10

answer : 10

텍스트, 폰트, 번호, 스크린샷이(가) 표시된 사진

자동 생성된 설명

10) 1 \* 2 \* 3 \* 4

correct answer : 24

answer : 24

텍스트, 폰트, 번호, 스크린샷이(가) 표시된 사진

자동 생성된 설명

11) 1 + 2 \* 3 + 4 \* 5

correct answer : 27

answer : 27

폰트, 텍스트, 상징, 그래픽이(가) 표시된 사진

자동 생성된 설명

12) 1.1 + 2.2 \* 3.3 + 4.4 \* 5.5

correct answer : 32.56

answer : 32.560001

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

13) 0 + 0  
correct answer : 0

answer : 0

텍스트, 폰트, 화이트, 블랙이(가) 표시된 사진

자동 생성된 설명

14) 93921 + 39281 \* 10  
correct answer : 486731

answer : 486731

텍스트, 폰트, 화이트, 그래픽이(가) 표시된 사진

자동 생성된 설명

15) 30293 \* 1039 + 10 \* 49201 + 5  
correct answer : 31966442

answer : 31966442

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

16) ( 3 \* 4 + 3) + 1 \* 5

correct answer : 20

answer : 20

폰트, 텍스트, 화이트, 그래픽이(가) 표시된 사진

자동 생성된 설명

]

17) (3.5 \* 8 + 1) + (1.4 \* 7.5 + 2) + 5

correct answer : 46.5

answer : 46.500000

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

18) + 1 2

correct answer : wrong

answer : wrong

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

action table error는 yyparse에서 i의 값이 0이 되었을 경우 발생한다.

현재 stack에 0이 들어가있고, PLUS는 257 이므로 action[0][1] = 0 이므로 해당 에러가 발생한다.

19) 3 \* + 4

correct answer : wrong

answer : wrong

폰트, 텍스트, 그래픽, 상징이(가) 표시된 사진

자동 생성된 설명

action[7][PLUS – 256] = 0 이므로 action table error가 발생한다.

20) 7 \* 8 +

correct answer : wrong

answer : wrong

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

action[6][END – 256] = 0 이므로 action table error가 발생한다.

20) 3 / 4

correct answer : wrong

answer : wrong

폰트, 텍스트, 그래픽, 로고이(가) 표시된 사진

자동 생성된 설명

yylex에서 정해진 토큰 외에 다른 토큰이 들어오면 해당 에러를 발생시킨다.

이 때 / 는 미리 정의되어 있지 않으므로 해당 에러가 발생하였다.

20) 3 / 4

correct answer : wrong

answer : wrong

텍스트, 폰트, 그래픽, 화이트이(가) 표시된 사진

자동 생성된 설명

실수 판정 중 . 이 한 번 더 나올 경우 해당 에러가 발생한다.

5..0 으로 .이 2번 나왔기에 해당 에러가 발생하였다.

4. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdbool.h>  #include <stdlib.h>  #include <ctype.h>  #define NUMBER 256  #define PLUS 257  #define STAR 258  #define RPAREN 260  #define END 261  #define LPAREN 259  #define EXPRESSION 0  #define TERM 1  #define FACTOR 2  #define ACC 999  // 과제 제출 화면 촬영시, terminal에서 진행할 것  struct TT {  bool isint; // int = true, float = false;  union a {  int i;  float f;  } value;  } typedef TT;  int action[12][6] = {  {5, 0, 0, 4, 0, 0}, {0, 6, 0, 0, 0 , ACC}, {0, -2, 7, 0, -2, -2},  {0, -4, -4, 0, -4, -4}, {5, 0, 0, 4, 0, 0}, {0, -6, -6, 0, -6, -6},  {5, 0, 0, 4, 0, 0}, {5, 0, 0, 4, 0, 0}, {0, 6, 0, 0, 11, 0},  {0, -1, 7, 0, -1, -1}, {0, -3, -3, 0, -3, -3}, {0, -5, -5, 0, -5, -5}  };  int go\_to[12][3] = {  {1,2,3},{0,0,0}, {0,0,0},{0,0,0},{8,2,3},{0,0,0},{0,9,3},{0,0,10},{0,0,0},{0,0,0},{0,0,0},{0,0,0}  };  int prod\_left[7] = {0, EXPRESSION, EXPRESSION, TERM, TERM, FACTOR, FACTOR};  int prod\_length[7] = {0,3,1,3,1,3,1};  int sym;  TT yylval;  bool check\_warn;  struct {  int stk[1000];  int top;  } typedef int\_stk;  struct {  TT stk[1000];  int top;  } typedef TT\_stk;  int\_stk state\_stk;  TT\_stk val\_stk;  void int\_push(int a);  void int\_pop(int count);  int int\_top();  void TT\_push(TT a);  void TT\_pop();  TT TT\_top();  TT TT\_plus(TT a, TT b);  TT TT\_star(TT a, TT b);  void init();  TT yyparse();  int yylex();  void shift(int i);  void reduce(int i);  void yyerror(char \*s);  void print\_TT(TT a);  TT yyparse();  void print\_warn();  int main() {  TT result = yyparse();  print\_TT(result);  return 0;  }  void int\_push(int a){  state\_stk.stk[++state\_stk.top] = a;  }  void int\_pop(int count){  state\_stk.top -= count;  }  int int\_top(){  return state\_stk.stk[state\_stk.top];  }  void TT\_push(TT a){  val\_stk.stk[++val\_stk.top] = a;  }  void TT\_pop(){  val\_stk.top--;  }  TT TT\_top(){  return val\_stk.stk[val\_stk.top];  }  TT TT\_plus(TT a, TT b){  if(a.isint && b.isint){  a.value.i += b.value.i;  return a;  }  else if(a.isint && !b.isint){  b.value.f += (float)a.value.i;  print\_warn();  return b;  }  else if(!a.isint && b.isint){  a.value.f += (float)b.value.i;  print\_warn();  return a;  }  else{  a.value.f += b.value.f;  return a;  }  }  TT TT\_star(TT a, TT b){  if(a.isint && b.isint){  a.value.i \*= b.value.i;  return a;  }  else if(a.isint && !b.isint){  b.value.f \*= (float)a.value.i;  print\_warn();  return b;  }  else if(!a.isint && b.isint){  a.value.f \*= (float)b.value.i;  print\_warn();  return a;  }  else{  a.value.f \*= b.value.f;  return a;  }  }  void init(){  state\_stk.top = -1;  val\_stk.top = -1;  check\_warn = true;  }  TT yyparse() {  int i;  init();  int\_push(0);  sym = yylex();  do {  i = action[int\_top()][sym-256];  if(i == ACC)  printf("success \n");  else if(i > 0) shift(i);  else if(i < 0) reduce(-i);  else yyerror("action table error");  } while(i != ACC);  return TT\_top();  }  void shift(int i){  int\_push(i);  sym = yylex();  }  void reduce(int i){ // 규칙 번호에 따라 다른 일을 수행해야함 연산이 필요한 부분이 있을 수 있음  int old\_top;  int rule\_length = prod\_length[i];  int\_pop(rule\_length);  old\_top = int\_top();  int\_push(go\_to[int\_top()][prod\_left[i]]);  if(i == 1){  TT tmp1, tmp2;  tmp1 = TT\_top();  TT\_pop();  tmp2 = TT\_top();  TT\_pop();  TT\_push(TT\_plus(tmp1, tmp2));  }  else if(i == 2){  ;  }  else if(i == 3){  TT tmp1, tmp2;  tmp1 = TT\_top();  TT\_pop();  tmp2 = TT\_top();  TT\_pop();  TT\_push(TT\_star(tmp1, tmp2));  }  else if(i == 4){  ;  }  else if(i == 5){  ;  }  else if(i == 6){  ;  }  else{  yyerror("parsing table error");  }  return;  }  void yyerror(char \*s){  printf("%s\n", s);  exit(1);  }  int yylex() {  static char ch = ' ';  int i = 0;  while(ch == ' ' || ch == '\t' || ch == '\n') ch = getchar();  if(isdigit(ch)){  char arr[1001];  int top = -1;  do {  arr[++top] = ch;  ch = getchar();  } while(isdigit(ch));  TT tmp;  if(ch == '.'){  do {  arr[++top] = ch;  ch = getchar();  } while(isdigit(ch));  if(ch == '.') yyerror("wrong token"); // 실수 판정 중에는 . 이 2개 이상 등장해선 안된다.  arr[++top] = '\0';  tmp.isint = false;  tmp.value.f = atof(arr);  }  else{  arr[++top] ='\0';  tmp.isint = true;  tmp.value.i = atoi(arr);  }  yylval = tmp;  TT\_push(yylval);  return NUMBER;  }  else if(ch == '+'){  ch = getchar();  return PLUS;  }  else if(ch == '\*'){  ch = getchar();  return STAR;  }  else if(ch == '('){  ch = getchar();  return LPAREN;  }  else if(ch == ')'){  ch = getchar();  return RPAREN;  }  else if(ch == EOF){  return END;  }  else{  yyerror("Undefined token");  }  }  void print\_TT(TT a){  if(a.isint){  printf("answer : %d\n",a.value.i);  }  else{  printf("answer : %f\n",a.value.f);  }  return;  }  void print\_warn(){  if(check\_warn){  check\_warn = false;  printf("Waring !! this expression is hybride expression of int and float!\n");  }  } |