**컴파일러설계 Project1 Report**



컴퓨터소프트웨어학부

2018007974 김서연

1. 컴파일 방법과 환경

- 주어진 Makefile을 이용해 make함

![텍스트이(가) 표시된 사진

자동 생성된 설명]()

- 환경: Ubuntu 16.04.7 / gcc 5.4.0

2. 구현 방법과 동작 방식

main.c

- NO\_PARSE와 TraceScan을 TRUE로 설정함

- main()에서의 출력을 C-minus에 맞게 적절히 변경함

globals.h

- MAXRESERVED를 6으로 재정의함: reserved word가 6개로 바뀌었기 때문

- reserved word의 목록을 수정함: IF, ELSE, WHILE, RETURN, INT, VOID

- special symbol의 목록을 수정함: PLUS(+), MINUS(-), TIMES(\*), OVER(/), LT(<), LE(<=), GT(>), GE(>=), EQ(==), NE(!=), ASSIGN(=), SEMI(;), COMMA(,), LPAREN((), RPREN()), LBRACE([), RBRACE(]), LCURLY({), RCURLY(})

utils.c

- printToken()을 reserved word와 special symbol에 맞게 수정함

scan.c

- StateType에 적절한 state를 추가함(INASSIGN은 사용되지 않으므로 삭제함)

- reservedWords도 적절히 수정함

- getToken()에서 input에 따른 state와 currentToken을 적절히 정의함

: 이 때 initial state는 START이고, 한 글자씩 읽어 변수 c에 저장함

① 현재 state가 START인 경우

→ c가 숫자라면 INNUM으로, 알파벳이라면 INID로 가고, c가 white space인 경우 아무것도 하지 않고 save를 FALSE로 바꿔 무시함

→ c가 ‘=’, ‘<’, ‘>’, ‘!’인 경우 문자 두 개로 구성된 연산자일 수도 있으므로 이를 확인하기 위해 적절한 state(각각 EQ, LT, GT, NE)로 가도록 함

→ ‘/’인 경우 나눗셈 연산인지 주석의 시작(/\*)인지 확인하도록 INOVER로 가는데 이 때 주석일 가능성이 있으므로 save를 FALSE로 바꿈

② 현재 state가 DONE인 경우

→ c가 special symbol이거나 ERROR이므로 currentToken만 적절히 설정함

③ 현재 state가 INNUM이나 INID인 경우

→ 다음에 들어온 input이 조건을 만족하는지를 확인해서 그렇지 않으면 읽은 문자를 되돌려놓고 save를 FALSE로 바꾼 후 state를 DONE으로 바꾸고 currentToken을 적절히 설정함

→ 이 때 identifier의 문자 뒤에 숫자가 올 수 있으므로 INID에서는 !isdigit(c)라는 조건을 추가함

④ 현재 state가 INEQ, INLT, INGT, INNE인 경우

→ 두 문자로 이뤄진 연산자 중 첫 번째 문자가 이미 들어온 상태이므로 두 번째 글자가 이어서 들어오는지 확인함

→ 두 번째 글자(=)가 들어오지 않았다면 한 문자로만 이뤄진 연산자인 것이므로 읽은 문자를 되돌리고 save를 FALSE로 바꾼 뒤 currentToken을 적절히 설정함

→ 두 번째 글자(=)가 들어왔다면 각각 그에 해당하는 currentToken을 설정함

→ 두 경우 모두 token 파악은 끝났으므로 state는 DONE으로 바꿈

⑤ 현재 state가 INOVER인 경우

→ 두 번째 글자(\*)가 들어오지 않았다면 나눗셈 연산인 것이므로 읽은 문자를 되돌리고 save를 FALSE로 바꾼 뒤 currentToken을 OVER로 설정함

→ 두 번째 글자(\*)가 들어왔다면 멀티라인 주석의 시작이므로 save를 FALSE로 바꾸고, 주석이 끝나는 것을 확인하기 위해 state는 INCOMMENT로 바꿈

⑥ 현재 state가 INCOMMENT인 경우

→ c는 ‘/\*’ 뒤에 오는 문자, 즉 주석이므로 저장하지 않음(save=FALSE)

→ 주석이 끝날 때 ‘\*/’가 나오므로 c가 ‘\*’이면 state를 INCOMMENT\_로 바꿔 해당 state에서 주석이 끝나는 것을 확인할 수 있도록 함

⑦ 현재 state가 INCOMMENT\_인 경우

→ INCOMMENT에서 ‘\*’을 확인하고 넘어온 state이므로 ‘/’가 나오면 주석이 끝남

→ 따라서 c가 ‘/’면 state를 START로 바꿔 다시 scan을 계속하도록 함

→ c가 EOF면 파일이 끝난 것이므로 state는 DONE으로, currentToken은 ENDFILE로 바꿈

Cminus.l

- definition section에서 identifier의 정규식을 수정함: 문자 뒤에 숫자가 나올 수 있도록 함

- rule section에서 주석을 제외한 special symbol과 reserved word들은 단순히 token을 반환하도록 함

- 이 때 읽은 정규식이 “/\*”라면 주석의 시작이므로, 주석의 끝인 “\*/”가 나올 때까지는 내용을 무시해야 함

- 따라서 char c가 ‘\*’이 아닌 동안 문자를 계속 읽고, ‘\*’를 만나면 그 다음에 다시 ‘/’를 만날 때까지 문자를 계속 읽는 단순작업을 수행함, 이 때 c가 EOF라면 파일이 끝났으므로 멈추고, ‘\n’이라면 정상적인 출력을 위해 lineno를 1 증가시킴

3. 예제 적용 결과

- 주어진 테스트 파일을 조금 변형하여 테스트함

![텍스트이(가) 표시된 사진

자동 생성된 설명]()

- 왼쪽이 cminus\_cimpl, 오른쪽이 cminus\_lex를 실행한 결과

![텍스트이(가) 표시된 사진

자동 생성된 설명]() ![텍스트이(가) 표시된 사진

자동 생성된 설명]()