

컴파일러 과제3 주요 QA – (사캠&PI)

1. 실습에서 썼던 `minic.l` 파일에는 숫자에 16진수가 같이 있는데 과제2에서는 16진수를 제외했었는데, 과제3에는 포함시키는게 맞는지
 - 과제3은 과제1, 2의 연장선입니다. 즉, 과제1의 심볼 테이블 관리 내용이 포함된 과제2의 `scanner.l`을 확장하여 과제3을 작성하는 것입니다. 기존 조건은 과제3에서도 동일하게 유지해주시기 바랍니다.
2. 해쉬테이블에 `id` 속성을 저장하는 것과 관련해서 예전 실습에서 진행한 `bubble.mc` 파일을 보면, `read()`; 나 `write()`;는 라이브러리 함수여서 함수 선언부가 없고 호출만 있는데, 이 경우에도 타입을 함수로 저장해주어야 하는지 아니면 타입이 선언된 적이 없다고 해주어야하는지 이런경우는 입력으로 들어오지 않는다고 생각해도 될지
 - 좋은 질문입니다. 그런데, 해쉬 심볼테이블 관리는 변수나 함수등이 선언/정의할 때만 하는 것이고, 사용할 땐, 저장할 필요가 없습니다.
이번학기에 배우는 것은 컴파일러를 100% 완성하는 것은 아니고, 문법체크까지만 하는데, 참고로, 만일 더 공부하게된다면, 정의하지 않은 변수의 사용이라던가 정의한지 않은 함수(라이브러리함수 제외) 사용의 경우 처리등도 하게 됩니다.
3. `factorial.mc` 파일에서 `factorial(n);`처럼 먼저 호출을 하고 나중에 함수 선언 및 정의 부분이 나올 경우, `line number`는 호출이 아닌, 선언 및 정의가 시작된 부분으로 하는 것이 더 맞는것인지
 - 현재 Mini-c (과제 3의 Modified Mini-C 역시)는 함수가 아무데에서나 정의되도 되는데, 즉 먼저 정의하고 사용(`call`)하거나, 사용먼저하고 뒤에 정의된 함수가 나옵니다. 모두 문법에 맞습니다. 그런데, 테스트할 땐, 함수가 사용하기 이전에 정의된 것으로만 테스트 할 예정입니다.
4. 교재의 `parser.y`는 보면 변수 선언을 할 때에 앞에 자료형 부분 문법이 사진처럼 되어있는데, 이렇게 되면 `const const int float x;` 처럼 앞에 `qualifer`와 `secifier`가 여러 개가 나타나도 `accept`하는 것 아닌지. 또, `void` 형 변수도 선언이 가능 등 문제가 많은 것 같은데 `miniC`에서는 이것이 허용되는 것인지, 아니면 저희가 이 부분들을 예러로 처리해주어야 하는 것인지.
 - Mini C 문법에서 과제 3를 위하여 변경하는 것은 과제 3 handout 및 관련강의노트에 설명한 것 밖에 없습니다. 그 이외는 책의 원래 Mini C와 동일하게 하면 됩니다. 즉 원래 책의 문법은 그대로 두라는 뜻입니다.

테스트 프로그램에선 상식적으로 우리가 C 프로그램 비슷한 정도로 구성하면 됩니다. 채점도 이렇게 할 예정입니다.

```

dcl_spec      : dcl_specifiers          { semantic(8); };
dcl_specifiers : dcl_specifier          { semantic(9); }
               | dcl_specifiers dcl_specifier { semantic(10);};
dcl_specifier  : type_qualifier         { semantic(11);}
               | type_specifier        { semantic(12);};
type_qualifier : tconst                 { semantic(13);};
type_specifier : tint                   { semantic(14);}
               | tvoid                  { semantic(15);};
function_name  : tident                 { semantic(16);};

```

5. 예러 처리를 하다보니 error 에서 빠져나오지 못하고 무한루프에 빠지는 경우가 종종 있었는데, 이 문제 관련하여 검색을 하다보니 yyclearin 이 현재 토큰을 버려주어서 이것을 활용하면 무한루프에서 빠져나올 수 있다는 것을 알게되었는데, 이 yyclearin 을 이번 과제 3 에서 활용해도 될지. 수업에서 다룬 적은 없는 것 같은데 사용해도 되는 것인지.

- 수업 때 가르쳐준 YACC 에 관한 것 이외에도 YACC 의 메뉴얼 (bison 메뉴얼)에 보면 훨씬 더 많은 풍부한 기능이 있는데, 이것을 활용하는 것은 매우 좋습니다. (LEX 도 마찬가지)

그런데, 문법 체크를 정교하게 하려는 나머지 error 를 너무 많이 삽입하여 문법을 고치게 되면 이것 때문에 오류가 날 수 있습니다. 그래서 error 체크는 적절히 해야합니다.

이러한 것들 때문에 컴파일러 프로그램을 하면 할 수록 계속 문제가 있어 시간을 많이 쓰게됩니다..

참고로, 이런 상황을 감안하여 나중 채점할 때는 오류가 복잡하거나 여러개 섞이도록 꼬지않고 비교적 단순한 문법 오류를 대상으로 할 예정입니다.

6. 예시로 참고하라고 주신 코드 중 wrong statement 오류가 어떤 경우일 때 나오는 오류인지

- 스스로 찾아야할 것 같습니다. 예시로 준 샘플 프로그램은 완벽한 것은 아닙니다. 전반적으로 파서가 어떻게 돌아가는지 보는 건지 하나하나 디테일하게 보면 위험할 수도 있습니다.

➤

7. (1)과제 파일의 채점 기준에서 오류가 있는 데이터의 경우 '적어도 하나의 Error 를 detect 하고 메시지를 source line 과 함께 출력하는가?' 라고 되어있는데 모든 error 를 정확하게 출력하지 않고 하나 이상의 error 만 출력해도 괜찮은 건지. 예를 들어 저희가 생각했을 때 5 가지의 error 가 있다고 판단하는 경우, 7 가지의 error 를 출력하거나 3 가지의 error 를 출력하게 되어도 괜찮은지.
 - (2) 오류처리할때 모든 문법 오류를 처리하라고 하셨는데 모든 문법 오류라는게 범위를 정확히 모르겠어서 정해진 범위가 있는지
 - (3) Parser 오류와 관련된 모든 오류를 처리하지 않으면 감점 대상인지
 - (4) 과제 내용이 모든 error를 가능한 한 출력하라고 되어있는데, 교수님께서 올려 주신 참고용 소스코드에 있는 정도로 출력하면 되는 것인지.

➤ 오류 개수를 집계하고 한개만 오류출력 이렇게 프로그램하면 오히려 더 힘이 들기 때문에 권장하지 않습니다.

채점기준에 적어도 하나만-이라고 한 이유는, 이것은 상황에 따라 어떤 팀은 5 개 오류가 실제 있는데, 모두 정확하게 하는 경우도 있고, 또 어떤 팀은 3 개 이렇게 다양할 수 있습니다. 이것은 비단 과제에서 뿐만아니라 실제 상용 컴파일러도 완벽하게 한번에 모든 오류를 detect / 원인을 알려주지 못할 수 있습니다. Error 가 났다고 하면 이때 주는 오류원인을 참고하여 고치고 다시 컴파일해서 만일 또다른 error 가 알려주고 또 고치고 이런식으로 실제합니다.

이번 과제에선 채점기준을 주어야 하니, 오류가 있는데 있다는 걸 전혀 메시지를 내주지 못하는 경우만 점수 깎겠다는 뜻입니다.
8. 오류가 있는 데이터의 경우 HStable 의 내용을 check 하지 않는다고 되어있는데 그러면 아예 출력하지 않아도 괜찮은지

➤ 출력해주시기 바랍니다. 참고로 HStable 의 내용은 나중에 조교가 오류없는 테스트프로그램으로 돌릴때 모두 체크할 예정입니다. 오류있는 프로그램에 대하여 HStable 체크안하는 이유는 두번 중복 채점할 필요가 없기때문이고, 더욱이 의미가 없기 때문입니다.
9. 과제 3 handout 에서, scanner.l 에서 ID 인 경우 character string 을 HStab 에 저장(즉, ST 의 인덱스를 저장)하고, parser.y 에서 ID type 을 좀더 세밀히 classify 하여 HStab 의 해당 ID 의 속성을 update 하라. (예, ID 인 경우 => integer scalar 변수, float scalar 변수, integer array 변수, float array 변수, void 함수명, integer 함수명, float 함수명, ...함수의 ..type 의 파라미터 등) – 라고 하셨는데, ... 함수의 ..type 의 파라미터 이 부분이 조금 이해가 안되는데, 어떤 속성을 출력해야 하는지.

- 예를 들어 `f(int x)`라고 하면 `x`를 보통 프로그래밍 언어 문법에서는 매개변수(파라미터)라고 부릅니다. 이런 경우, `x`는 함수 `f`의 integer scalar parameter variable 입니다.

10. 그러면, 예를 들어 'factorial.mc'의 main 에 있는 integer scalar variable `n` 과 factorial 의 파라미터로 있는 integer parameter scalar variable `n` 처럼 중복되는 변수이름을 갖고 있더라도 모두 속성을 따로 출력해주어야 하는지.

- 상식적으로 프로그래밍 언어 문법 측면에서 생각하면 다른 변수입니다. 같은 이름이지만 scope 가 다른 다른 변수입니다. 이것을 처리하려면 심볼테이블에서 관리할 때 추가적인 작업이 필요하게 됩니다. 그래서 이번 과제에서는 채점할 때 같은 변수 이름인데, scope 가 다른 것은 테스트데이터에 없도록 할 예정입니다. 따라서 이 문제에 대한 처리는 해도 좋고 안해도 좋습니다.

11. (1) 이번 과제는 보고서는 없고 zip 파일만 내면 되는 게 맞는지

(2) 입출력 파일 부분에서, 팀별로 정의한 내용을 기술해야 하는 부분은 어디에 기술하면 될지. 보고서 형식으로 제출해야 하는 건지.

- (1) 전체 소스코드(main.c, parser.y, scanner.l, symtable.c, reporterror.c, tn.g, glob.h 등//lex.yy.c, parser.tab.c 도 반드시 제출)와 (2)입출력 내용(문법 오류가 없는 것 4 개 이상, 문법 오류가 있는 것 4 개 이상), (3)기여도를 각각 zip 하여 제출해주시기 바랍니다. 입출력 내용과 기여도는 캡처를 해서 제출해도 되고 보고서 형식으로 작성해서 내셔도 됩니다. *output 의 경우 과제물에 명확히 정의되지 않은 부분이 있어 본인이 정의한 것이 있을 경우 그 내용을 기술하고 해당 output 부분에 표시-내용 기술 시 형식은 자유(캡처, 워드, txt 파일 등)

12. identifier 타입이 함수명일때 리턴타입을 저장하기 위해 identifier 속성에 int returnType 을 추가해도 되는지

- 안됩니다. C 문법에서 생각해보면 int return type 이라는 건 없습니다. 예를들어, 문법에서 선언하는 부분만 찾아 들어가서 해야하고, 지금 함수를 처리하는 상황이라는 걸 알게 되면, 그때 int 를 해시테이블로 처리하고, 출력은 리턴 타입이라고만 출력해야 합니다. intreturnType 이란 타입이 있지는 않습니다. (해시테이블의 구조를 바꿔도 되긴 하지만, 프로그램 구현 디테일이라 더 성가실 것 같습니다.)

13. reporterror 파일에서 scanner 와 parser 의 오류를 출력하게끔 되어있는데 scanner 오류를 정의한 enum errortype 안에 parser 오류를 추가해야하는지, 아님 따로 parser 오류를 정의해도 되는지.

➤ 팀에서 알아서 정의하면 됩니다.

14. reporterror 에서 yyerror(char *s)를 꼭 사용해야 하는지.

➤ 사용해야 합니다. 지금까지 과제/실습하면서 해당 부분에 문제가 없었나요?