Design and Development of Compiler  
for C- Language

Phase 2: Design and Implementation of LALR Parser

1. Project2 제안서

과목명: [CSE4120] 기초 컴파일러 구성

담당교수: 서강대학교 컴퓨터공학과 정 성 원

개발자: 20101631 모지수

개발기간: 2015. 05. 05 - 2015. 05. 14

Project2 제 안 서

프로젝트 제목: Design and Development of Compiler for C- Language:

Phase 2: Design and Implementation of LALR Parser

제출일: 2015. 05. 07.

개발자: 모지수

I. 개발 목표

- 이 프로젝트는 C- Parser를 구현하는 것을 목표로 한다.

- Parser는 project1에서 구현한 lexical analyzer와 다음과 같이 실시간으로 상호작용한다. Parser는 lexical analyzer가 output으로 내놓는 token을 받아서 이들을 (이 경우 C- 의) grammar에 맞게 해석하여 abstract syntax tree로 구성하여 output으로 내놓게 된다.



II. 개발 범위 및 내용

- 자신이 설계한 개발 목표를 달성하기 위하여 어떠한 내용의 개발을 수행할 지 그 범위

와 예상되는 개발 내용을 기술할 것.

가. 개발 범위  
Yacc/Bison을 통해 컴파일 할 C- Language의 Syntax listing(in .y file), global.h(Global types and vars for C- compiler), scan.h(The scanner interface for the C-compiler), util.h(Utility function interface for the C- compiler), util.c(Utility function implementation for the C- compiler)

나. 개발 내용

- 기존 코드의 syntax tree 선언부에 추가/변경할 사항이 있다.

NodeKind : typedef enum {stmtK, Expk, Declk} NodeKind;  
StmtKind : typedef enum {IfK, elseK, WhileK, ReturnK} StmtKind;  
 ExpKind : typedef enum {OpK, ConstK, IdK, CallKind, ArrayKind} ExpKind;  
DeclKind : typedef enum {VarK, FunK, ParamK} DeclKind;  
ExpType : typedef enum {Void, Int} ExpType;

- BNF Grammar for C-Minus

1. *program* ***→*** *declaration-list*
2. *declaration-list* ***→*** *declaration-list declaration |* declaration
3. declaration***→*** *var-declaration |* fun-declaration
4. *var-declaration* ***→*** *type-specifier* **ID****;** *| type-specifier* **ID [ NUM ] ;**
5. *type-specifier* ***→*****int** *|* **void**
6. fun-declaration***→*** *type-specifier* **ID****(** *params* **)**compound-stmt
7. *params* ***→*** *param-list |* **void**
8. *param-list* ***→*** *param-list* **,** *param |* param
9. *param* ***→*** *type-specifier* **ID** *| type-specifier* **ID [ ]**
10. compound-stmt***→*****{** *local-declarations statement-list* **}**
11. local-declarations***→*** *local-declarations var-declarations |* empty
12. statement-list***→*** *statement-list statement |* empty
13. statement***→*** *expression-stmt | compound-stmt | selection-stmt | iteration-stmt |* return-stmt
14. expression-stmt***→*** *expression* **;** *|* **;**
15. *selection-stmt* ***→*****if (** *expression* **)** *statement |* **if (** *expression* **)** *statement* **else**statement
16. *iteration-stmt* ***→*****while (** *expression* **)** *statement*
17. *return-stmt* ***→*****return ;** *|* **return** *expression* **;**
18. expression***→*** *var* **=** *expression |* simple-expression
19. var***→*****ID** *|* **ID****[** *expression* **]**
20. simple-expression***→*** *additive-expression relop additive-expression |* additive-expression
21. relop***→*****<=** *|* **<** *|* **>** *|* **>=** *|* **==** *|* **!=**
22. additive-expression***→*** *additive-expression addop term |* term
23. addop***→*****+** *|* **-**
24. term***→*** *term mulop factor |* factor
25. mulop***→*****\*** *|* **/**
26. factor***→*****(** *expression* **)** *| var | call |* **NUM**
27. call***→*****ID (** *args* **)**
28. args***→*** *arg-list |* empty
29. *arg-list* ***→*** *arg-list* **,** *expression |* expression

Keywords: **else if int return void while**

Special symbols: **+ - \* / < <= > >= == != = ; , ( ) [ ] { } /\* \*/**

**ID = letter letter\***

**NUM = digit digit\***

**letter = a | .. | z | A | .. | Z**

**digit = 0 | .. | 9**

Comments: **/\* ... \*/**

III. 추진 일정 및 개발 방법

가. 추진 일정

- 5/6~5/7 : 개발 내용/방법 방향잡기, 제안서 작성  
- 5/5~5/12 : 개발 내용 구현 및 디버깅  
- 5/13~5/14 : 결과보고서 작성

나. 개발 방법

- project 1에서 구현한 .l file 의 선언 부분을 주석처리 후 진행하도록 한다.

IV. 기타

- Yacc/Bison 관련 매뉴얼 : http://www.gnu.org/software/bison/manual/bison.html