情報工学実験Cレポート
コンパイラ実験

氏名: 島谷　隼生 (Shimatani, Toshiki)
学生番号: 09428526

出題日: 2018 年 12 月 06 日
提出日: 2019 年 02 月 05 日
締切日: 2019 年 02 月 05 日

1　実験の目的

本実験では，情報系学科で学んだ 3 年間の総仕上げとしてコンパイラの作成を行うことを通じ，プログラミング言語で書かれたプログラムとアセンブリ言語の対応についてより深く理解し，木構造の取扱いについて習熟すること，および yacc.lex といったプログラムジェネレータを使用してプログラムを作成する経験を積むことを目的とする．

2　作成した言語定義

最終的に作成した言語の定義を BNF で記述する．以下がその言語定義である．

作成した言語定義 (1/3

$$
\begin{aligned}
< program > &::= < variable\_declarations >< function\_list > \\
< function\_list > &::= < function >< function\_list > \\
&\mid\ < function > \\
< function\_list > &::= < function >< function\_list > \\
< function > &::= < pre\_func > (< argument\_list >) \\
&\quad < variable\_declarations >< statement\_list > \\
&\mid\ < pre\_func > () \\
&\quad \{< variable\_declarations >< statement\_list >\} \\
< pre\_func > &::= \texttt{func} < \texttt{IDENTIFIER} > \\
< argument\_list > &::= < argument > , < argument\_list > \\
&\mid\ < argument > \\
< argument > &::= \texttt{define} < \texttt{IDENTIFIER} > \\
&\mid\ \texttt{array} < \texttt{IDENTIFIER} > [] \\
&\mid\ \texttt{array} < \texttt{IDENTIFIER} > [< \texttt{NUMBER} >][] \\
< variable\_declarations > &::= < declaration >< variable\_declarations >
\end{aligned}
$$

作成した言語定義 (2/3)

$$
\begin{array}{rcl}
< declaration > & ::= & \texttt{define} < identifier\_list > \texttt{;} \\
& | & \texttt{array} < \texttt{IDENTIFIER} > [< \texttt{NUMBER} >]\texttt{;} \\
& | & \texttt{array} < \texttt{IDENTIFIER} > [< \texttt{NUMBER} >][< \texttt{NUMBER} >]\texttt{;} \\
< identifier\_list > & ::= & < \texttt{IDENTIFIER} > \texttt{,} < identifier\_list > \\
& | & < \texttt{IDENTIFIER} > \\
< statement\_list > & ::= & < statement >< statement\_list > \\
& | & < statement > \\
< statement > & ::= & < assignment\_statement > \\
& | & < loop\_statement > \\
& | & < selection\_statement > \\
& | & < function\_call > \\
& | & < break\_statement > \\
& | & < \texttt{IDENTIFIER} >< unary\_operator > \\
& | & < unary\_operator >< \texttt{IDENTIFIER} > \\
< assignment\_statement > & ::= & < \texttt{IDENTIFIER} > \texttt{=} < arithmetic\_expression > \texttt{;} \\
& | & < array\_reference > \texttt{=} < arithmetic\_expression > \texttt{;} \\
< arithmetic\_expression > & ::= & < arithmetic\_expression >< additive\_operator > \\
& & < multiplicative\_expression > \\
& | & < multiplicative\_expression > \\
< multiplicative\_expression > & ::= & < multiplicative\_expression > \\
& & < multiplicative\_operator >< primary\_expression > \\
& | & < primary\_expression > \\
primary\_expression & ::= & < variable > \\
& | & (< arithmetic\_expression >) \\
< additive\_operator > & ::= & \texttt{+} \\
& | & \texttt{-} \\
< multiplicative\_operator > & ::= & \texttt{*} \\
& | & \texttt{/} \\
& | & \texttt{\%} \\
< unary\_operator > & ::= & \texttt{++} \\
& | & \texttt{--} \\
< variable > & ::= & \texttt{IDENTIFIER} \\
& | & \texttt{NUMBER} \\
& | & < array\_reference > \\
& | & \texttt{IDENTIFIER} < unary\_operator > \\
& | & < unary\_operator > \texttt{IDENTIFIER} \\
< array\_reference > & ::= & \texttt{IDENTIFIER}[< variable >] \\
& | & \texttt{IDENTIFIER}[< variable >][< variable >] \\
< loop\_statement > & ::= & \texttt{while}(< expression >)< statement\_list > \\
& | & \texttt{WHILE}(< expression >) < statement > \\
& | & \texttt{FOR} \\
& & (< for\_initial >< for\_expression >< for\_update >) \\
& & < statement\_list > \\
& | & \texttt{FOR} \\
& & (< for\_initial >< for\_expression >< for\_update >) \\
& & < statement >
\end{array}
$$

作成した言語定義 (3/3)

$$
\begin{array}{rcl}
< for\_initial > & ::= & < assignment\_statement > \\
 & | & \texttt{;} \\
< for\_expression > & ::= & < expression > \texttt{;} \\
 & | & \texttt{;} \\
< for\_update > & ::= & \texttt{<IDENTIFIER>=}< arithmetic\_expression > \\
 & | & < array\_reference >=< arithmetic\_expression > \\
 & | & \texttt{IDENTIFIER}< unary\_operator > \\
 & | & < unary\_operator >\texttt{<IDENTIFIER>} \\
 & | & \text{(何もないことを定義)} \\
< selection\_statement > & ::= & < if\_statement > \\
 & | & < if\_statement >< else\_statement > \\
< if\_statement > & ::= & \texttt{IF(}< expression >\texttt{)}< statement\_list > \\
 & | & \texttt{IF(}< expression >\texttt{)} < statement > \\
< else\_statement > & ::= & \texttt{ELSE}< statement\_list > \\
 & | & \texttt{ELSE} < statement > \\
< break\_statement > & ::= & \texttt{BREAK;} \\
< expression > & ::= & < arithmetic\_expression >< comparison\_operator > \\
 & & < arithmetic\_expression > \\
< comparison\_operator > & ::= & \texttt{==} \\
 & | & \texttt{<} \\
 & | & \texttt{>} \\
 & | & \leqq \\
 & | & \geqq \\
< function\_call > & ::= & \texttt{FUNCCALLIDENTIFIER();} \\
 & | & \texttt{FUNCCALLIDENTIFIER(}< parameter\_list >\texttt{);} \\
< parameter\_list > & ::= & < arithmetic\_expression > \texttt{,} < parameter\_list > \\
 & | & < arithmetic\_expression >
\end{array}
$$

3 言語定義で受理されるプログラムの例

4 コード 生成の概略

5 コンパイラ作成過程で工夫した点

6 最終課題を解くために書いたプログラムの概要

7 最終課題の実行結果

8 最終課題の考察

9 ソースプログラムのある場所

作成したソースプログラ以下の場所に保存してある..

10 プログラムリスト

```
<program> ::= <variable_declarations> <function_list>
            | <function_list>
<function_list> ::= <function> <function_list>
                  | <function>
<function> ::= <pre_func> (<argument_list>) {<variable_declarations> <statement_list>}
             | <pre_func> () {<variable_declarations> <statement_list>}
<pre_func> ::= func <IDENTIFIER>
<argument_list> ::= <argument>, <argument_list>
                  | <argument>
<argument> ::= define <IDENTIFIER>
             | array <IDENTIFIER>[]
             | array <IDENTIFIER> [<NUMBER>][]
<variable_declarations> ::= <declaration> <variable_declarations>
<declaration> ::= define <identifier_list>;
                | array <IDENTIFIER> [<NUMBER>];
                | array <IDENTIFIER> [<NUMBER>][<NUMBER>];
<identifier_list> ::= <IDENTIFIER>, <identifier_list>
                    | <IDENTIFIER>
<statement_list> ::= <statement> <statement_list>
                   | <statement>
<statement> ::= <assignment_statement>
              | <loop_statement>
              | <selection_statement>
              | <function_call>
              | <break_statement>
              | <IDENTIFIER> <unary_operator>
              | <unary_operator> <IDENTIFIER>
<assignment_statement> ::= <IDENTIFIER> = <arithmetic_expression>;
                         | <array_reference> = <arithmetic_expression>;
<arithmetic_expression> ::= <arithmetic_expression> <additive_operator> <multiplicative_expression>
                          | <multiplicative_expression>
<multiplicative_expression> ::= <multiplicative_expression> <multiplicative_operator> <primary_expres
                              | <primary_expression>
primary_expression : variable
| PAREN_L arithmetic_expression PAREN_R
additive_operator : ADDITION
| SUBTRACTION {$$ = AST_SUB;}
multiplicative_operator : MULTIPLICATION
| DIVISION
| MODULUS
unary_operator : INCREMENT
| DECREMENT
variable : IDENTIFIER
| NUMBER
| array_reference
| IDENTIFIER unary_operator
| unary_operator IDENTIFIER
array_reference : IDENTIFIER BRACKET_L variable BRACKET_R
| IDENTIFIER BRACKET_L variable BRACKET_R BRACKET_L variable BRACKET_R
loop_statement : WHILE PAREN_L expression PAREN_R BRACE_L statement_list BRACE_R
| WHILE PAREN_L expression PAREN_R statement
| FOR PAREN_L for_initial for_expression for_update PAREN_R BRACE_L statement_list BRACE_R
| FOR PAREN_L for_initial for_expression for_update PAREN_R statement
for_initial : assignment_statement
```