

2018 编译原理课程实践 1

一. 简化版的 C 语言声明的文法产生式如下:

translation_unit

```
: declaration
| translation_unit declaration
;
```

declaration

```
: declaration\_specifiers init\_declarator\_list ';'
;
```

declaration_specifiers

```
: type\_specifier
;
```

init_declarator_list

```
: init\_declarator
| init_declarator_list ',' init\_declarator
;
```

init_declarator

```
: declarator
;
```

type_specifier

```
: VOID
| INT
;
```

declarator

```
: pointer direct\_declarator
| direct\_declarator
;
```

direct_declarator

```
: IDENTIFIER
| '(' declarator ')'
| direct_declarator '[' CONSTANT\_INT ']'
| direct_declarator '(' parameter\_type\_list ')'
| direct_declarator '(' ')'
;
```

```

pointer
: '*'
| '*' pointer
;

```

```

parameter_type_list
: parameter list
;

```

```

parameter_list
: parameter declaration
| parameter list ',' parameter declaration
;

```

```

parameter_declaration
: declaration_specifiers declarator
;

```

二. 针对上述声明文法，实现一个递归下降的分析处理程序。该程序可以：

- (1) 能够将输入中各个声明变量的类型表达式等相关信息加以输出；
- (2) 可进行简单的类型检查，如函数返回值类型不能是数组，而数组的元素类型也不能是函数类型等。
- (3) 可输出相关类型的大小。假设 `int` 与指针类型大小均为 1，那么 `int a[20]` 的大小为 20。

参考示例如下：

输入文件 `test.pl`:

```

int *p;

int *pa[20][30];

int (**pp);

int (**ptr[20])[30];

int (*f(int i,int *j))[20];

int * fa(int i)[20];

int af[20](int k);

```

输出信息可以是:

Starting parsing variable declaration...

```
int *p;
p is type of : pointer(int)
Type Chencking ... OK! type size : 1
```

ValueType is : int

```
int *pa[20][30];
pa is type of : array(20,array(30,pointer(int)))
Type Chencking ... OK! type size : 600
```

ValueType is : array(30,pointer(int))

```
int (**pp);
pp is type of : pointer(pointer(int))
Type Chencking ... OK! type size : 1
```

ValueType is : pointer(int)

```
int (**ptr[20])[30];
ptr is type of : array(20,pointer(pointer(array(30,int))))
Type Chencking ... OK! type size : 20
```

ValueType is : pointer(pointer(array(30,int)))

```
int (*f(int i,int *j))[20];
parameter: i is type of : int
parameter: j is type of : pointer(int)
f is type of : function(int X pointer(int) =>
pointer(array(20,int)))
Type Chencking ... OK!
Para-Type is : int X pointer(int), Return-Type is :
pointer(array(20,int))
```

```
int * fa(int i)[20];
parameter: i is type of : int
fa is type of : function(int => array(20,pointer(int)))
Type Chencking ... ^
Error XXX: Array or Function can not be retured from functions.
error!
```

```

int af[20](int k);
af is type of : array(20,function(int => int))
Type Chencking ...                ^
Error XXX: Array of functions is not allowed.
error!

```

- 三. 可以借助 PLO 编译程序实现中的词法分析和相关数据结构之设定来完成你的类型分析程序。