实验二 语义分析

任课老师：戴新宇

助教：

尚迪(shangd@nlp.nju.edu.cn)

胡光能(hugn@nlp.nju.edu.cn)

概要

* 实验1总结
* 实验2任务（必做 + 选做）
* 实验2讲解
  * 整体思路
  * 错误类型1如何解决
  * 数据结构如何设计
* 实验2会遇到的问题

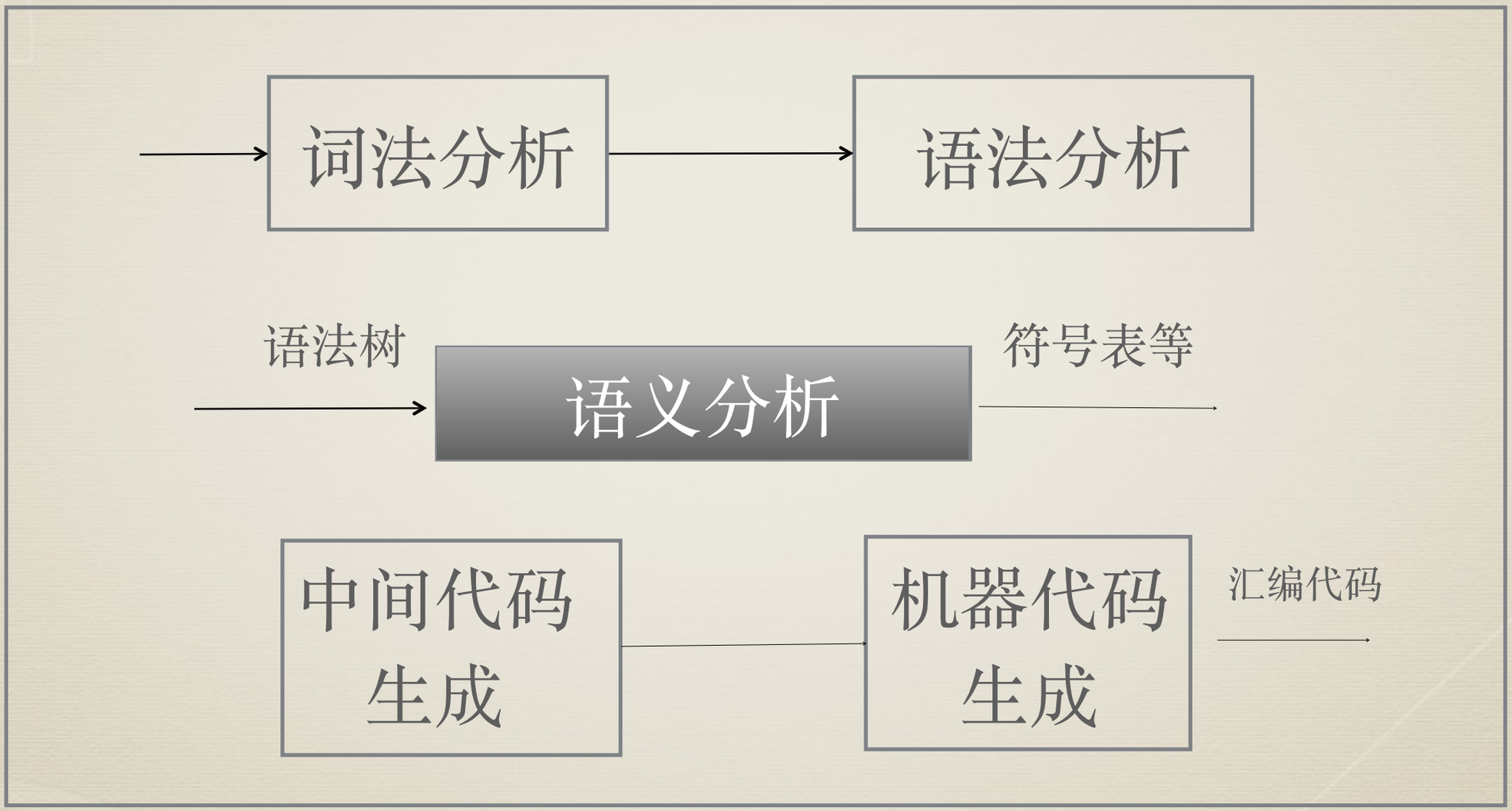
实验1总结

* 务必提交所有的代码文件
  * .l .y .h .c 和可执行程序(parser)
  * 报错信息务必不要使用中文

* 实验1选做内容
  * 没有完成选做内容不会影响后续的实验

* 实验1必做内容
  * 没有完成必做内容后续实验是无法继续进行的

编译器模块分解图



实验任务

* 必做内容
  * 错误类型1：变量在使用时未经定义
  * 错误类型2：函数在调用时未经定义
  * 错误类型3：变量经过重复定义
  * 错误类型4：函数经过重复定义
  * 错误类型5：赋值号两边表达式类型不匹配
  * 错误类型6：赋值号左边出现了一个只有右值的表达式

    ...

  错误类型12：数组访问操作符[ ]中出现非整数

实验任务

* 选做内容
  * 错误类型13：对非结构体型变量使用“ . ”操作符
  * 错误类型14：访问结构体中未定义过的域
  * …
  * 错误类型18：函数进行了声明，但没有被定义
  * 错误类型19：函数的多次声明互相冲突

* 报错格式：
  * Error type [错误类型] at line [行号]: [说明文字]

实验任务

* 对语法树进行语义分析和类型检查
  * 输入文件无任何词法和语法错误
  * 无任何实验一的选做内容

* 语义分析
  * 在整个流程中比较简单，不需要依靠工具，手写代码实现

* 基本要求
  * 需要精心设计符号表和变量类型的数据结构
  * 需要安全合理地维护你的语法树
  * 需要熟练使用C语言

整体思路

* SDT，在bison的文件里嵌入语义分析的代码
  * 优点：实现较简单
  * 缺点：所有代码都在bison文件里
* 先建立语法树再遍历语法树执行分析（推荐）
  * 优点：层次性好
  * 缺点：实现起来比第一种方法复杂

整体思路

* 在可能会产生实验要求里给出的错误的地方加入检查代码
  * 如果采用第一种方法，就在相应处嵌入检查代码
  * 如果采用第二种方法，就在遍历到相应的节点时，检查它的子树

整体思路

*至少维护三个列表（插入，查找和删除）

* 符号表
* Struct类型表
* 函数类型表
* 若要完成选做部分，还需维护更多的信息

如何解决错误类型1

* 错误类型1:
* 变量在使用时未经定义

```
int main(int a)
{
    int i = 0;
    z = i+a;
    return 0;
}
```

* Error type 1 at line 4: Undefined variable “z”

如何解决错误类型1

* 错误类型1：变量在使用时未经定义

深度遍历语法树

发现ExtDef节点

子节点是FunDec

后续节点是CompSt，表明正在进行函数定义

符号表和函数表加入函数定义

后续节点是SEMI，表明正在进行函数声明

符号表和函数表加入函数声明

子节点是ExtDecList

符号表中加入变量

发现Exp节点

子节点为ID

提取ID节点变量名对比符号表判断是否报错

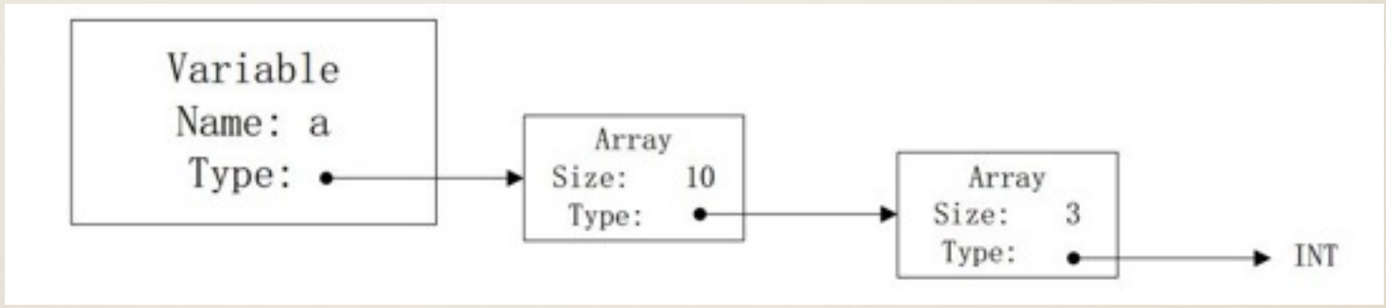
如何设计数据结构

* 数据结构：符号表的记录结构
  * name　：变量名函数名
  * funcOrVariable ： 变量或函数
  * visitedTag　：判断是定义还是声明
  * lineNumber　　：符号对应的行号
  * FunctionMessage ： 函数信息：返回值类型，函数参数类型，函数作用域标识等
  * VariableMesage ： 变量信息：类型(基本类型，数组类型，结构体类型)等
  * 指向下一条记录的指针

如何设计数据结构

* 一个简单的实现
  * N维数组的每个元素的类型是N-1维数组

int [10][3]

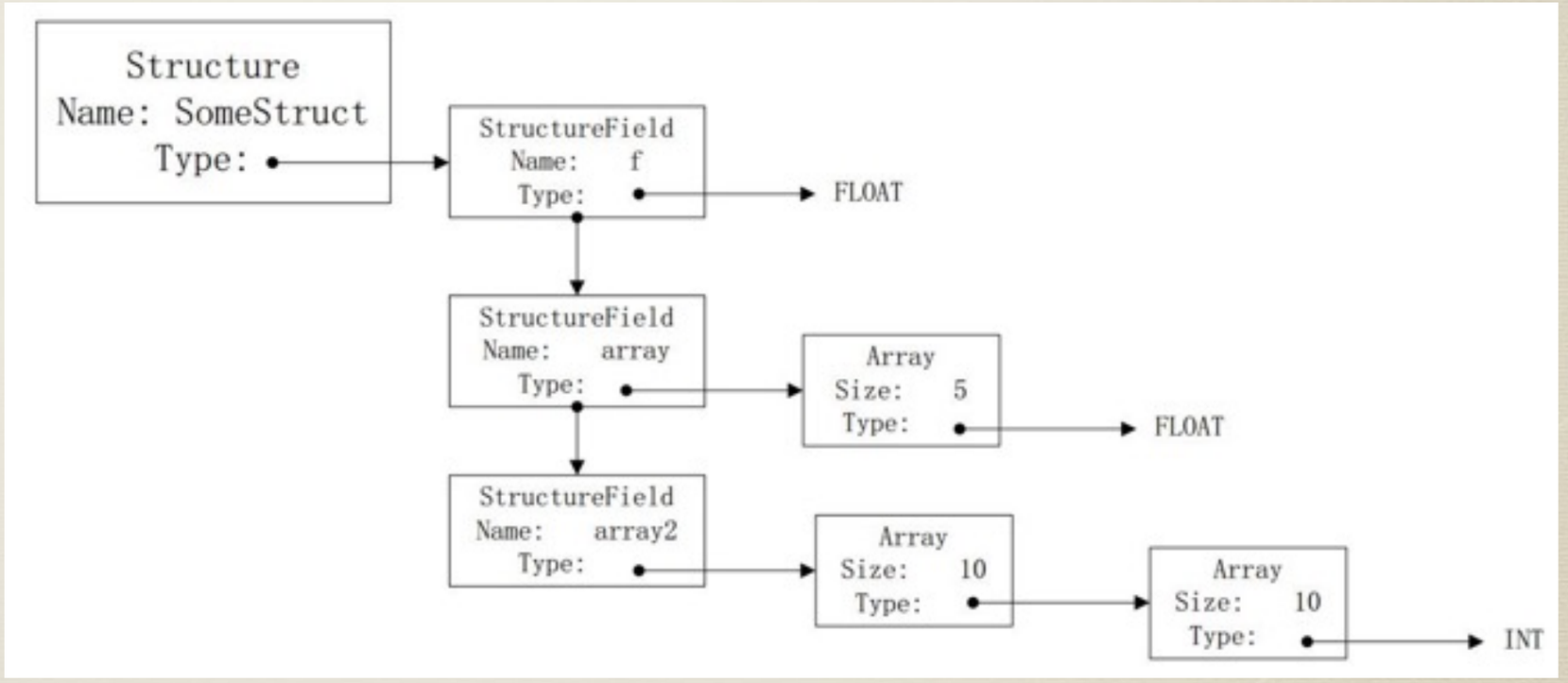


* 其它实现亦可（不必局限于我们介绍的这种）

保存结构体信息

* 思路和数组的类似，使用链表实现

```
struct SomeStruct
{
    float f;
    float array[5];
    int array2[10][10];
};
```



符号表的存储

* 链表
  * 优点：实现简单
  * 缺点：查找效率不高

* 平衡二叉树 （AVL,红黑树）
  * 优点：查找效率较高
  * 缺点：实现起来复杂（可从网上找代码）

* 哈希表
  * 优点：查询效率高
  * 缺点：存储需求大

嵌套作用域

* 基于栈的实现
  * 遇到大括号{即将当前符号表信息压栈
  * 遇到大括号}号时，删除当前符号表信息，
    * 从栈里pop出一个符号表信息
* 检查重复定义
  * 检查当前符号表，则从栈里查找（从栈顶到栈尾）
  * 返回第一个查找到的，如果都找不到则返回无定义错误

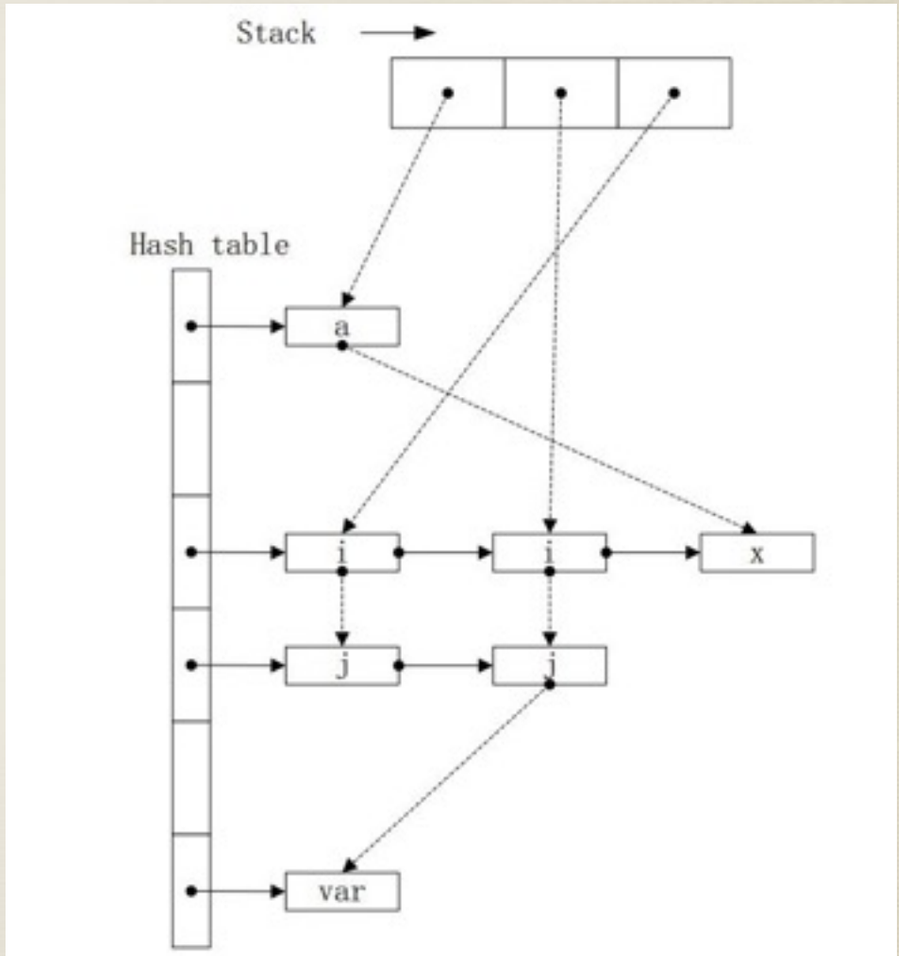
```
int i = 0;
int main(int a)
{
    int j = 0;
}
int test( int b)
{
    int c= i+j+b;
}
```

嵌套作用域

* 只使用一个符号表信息
  * 每次查找只返回最近的定义
  * 遇到}后需删除从{开始插入的符号表
  * (此方法强烈不推荐，存在安全隐患)

基于链表和哈希表的实现

- 使用哈希表存储符号表
- 栈是用来记录符号表的先后关系
- 同一个链表里是{语句插入的符号表



基于结构体的选作内容

* 加入相应的检查代码即可
  * e.g 结构体等价，逐个匹配每个成员的类型

如何输出错误信息

* 检查出所有错误，切勿遇到一个错误就退出
  * 每个测试样例中包含多钟类型的错误
* 运行完语义分析后，输出发现的所有错误
  * 可以检测到一个就输出一个
  * 也可以检测到一个就把错误信息存到链表里，最后遍历链表输出所有错误信息
  * 错误信息需包括：错误内容、行号、列号
  * 严格按照报错格式输出　Error type 1 at line 4: Undefined variable “z”

实验2可能出现的问题

* 段错误
  * 大多数段错误发生在树节点作为参数传入时未被定义过，直接使用
  * 使用参数前，判断是否为NULL
* Struct定义
  * 结构体的定义最好写在.h文件中
  * .y 和 .l头部声明部分保持精简

提交说明

* 地址：ftp://114.212.190.181: 21
* 用户名和密码：upload
* 格式：学号命名的压缩包 (zip/rar) 121220000_lab2.rar
* 内容：
  * 源程序(必须能通过编译)
  * 可执行程序(命名为 parser)
  * 报告PDF(完成的功能，编译步骤，实现方法，结点的数据结构表示；不超过3页)
* Deadline 大约在5月6日左右

Warning！！！

* 抄袭检测极其严格
* 请各位不要抱有侥幸心里

Thank you.
Any questions?

文法： S→ L=R|R

L→ *R|id

R→ L

1.请分别构造该文法的 LR(0) 和 LR(1) 相集规范族及自动机

2.构造 SLR 语法分析表和 LR(1) 分析表

3.基于分析表，写出串 id=id 和 *id=id 的移进规约的分析过程。