

MINI_C语言语义分析器

CC's Semantic Analyzer

语义分析
错误提示
用户体验

1113730116 陈晨

语法分析支持内容说明

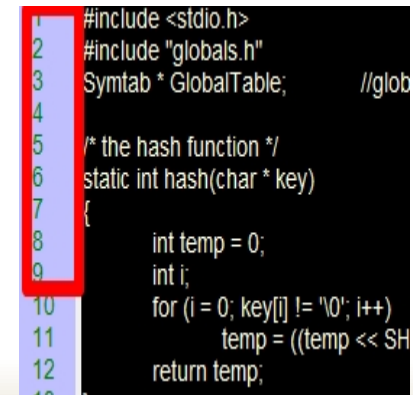
- 程序由多个函数组成，支持多个函数声明
- 函数定义支持指针以及基本类型定义
- 所有变量声明在函数体前端声明，支持多个变量声明
- 变量声明以及赋值支持基本类型，数组
- 支持基本算术表达式（+ - * /）符合优先结合律，
- 支持关系表达式<, >, >=, <=, ==, !=
- 支持if-else语句
- 支持while () { } 循环语句
- 支持数组赋值和引用

语义分析阶段错误提示

1. 类型不同赋值： `int` 赋值给 `float` 给出警告（合理）
`Float` 赋值给 `int` 报错。
2. 声明重复：符号表不添加重名变量,参数。
（考虑到为了提高实验容错率，函数名重名会报错，但依然在全局符号表出现）
3. 变量引用不存在：出现未声明就引用的符号时会报错
4. 数组访问出错：包括非数组类型引用下标，如 `int a`，引用：`a[12]`
下标越界，如 `int a[12]` 引用 `a[12]`
引用类型超界 `int a [12]` 引用 `a[11][13]`
5. 数组下标类型错误 例如 `a[c]=12`; 其中 `c` 是 `float` 型。

用户体验

- 支持文件读取，保存。
- 文本编辑框行号动态显示
- 支持错误快速定位
- 支持词法分析，语法分析
- 分析表可在程序运行后自动生成 *expression.txt*



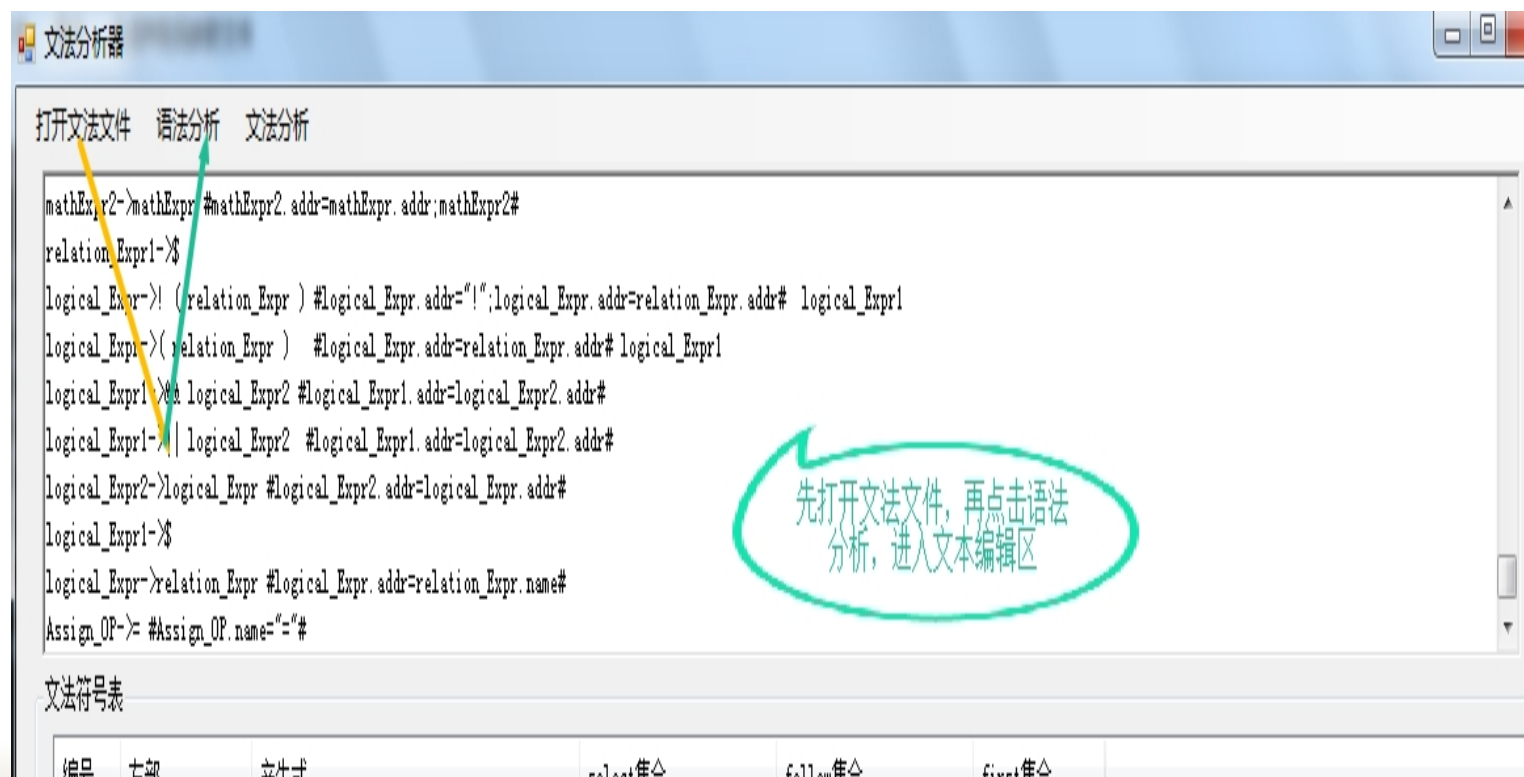
```
1 #include <stdio.h>
2 #include "globals.h"
3 Syntab * GlobalTable; //glob
4
5 /* the hash function */
6 static int hash(char * key)
7 {
8     int temp = 0;
9     int i;
10     for (i = 0; key[i] != '\0'; i++)
11         temp = ((temp << SH
12     return temp;
```

MINI_C语言语义分析器

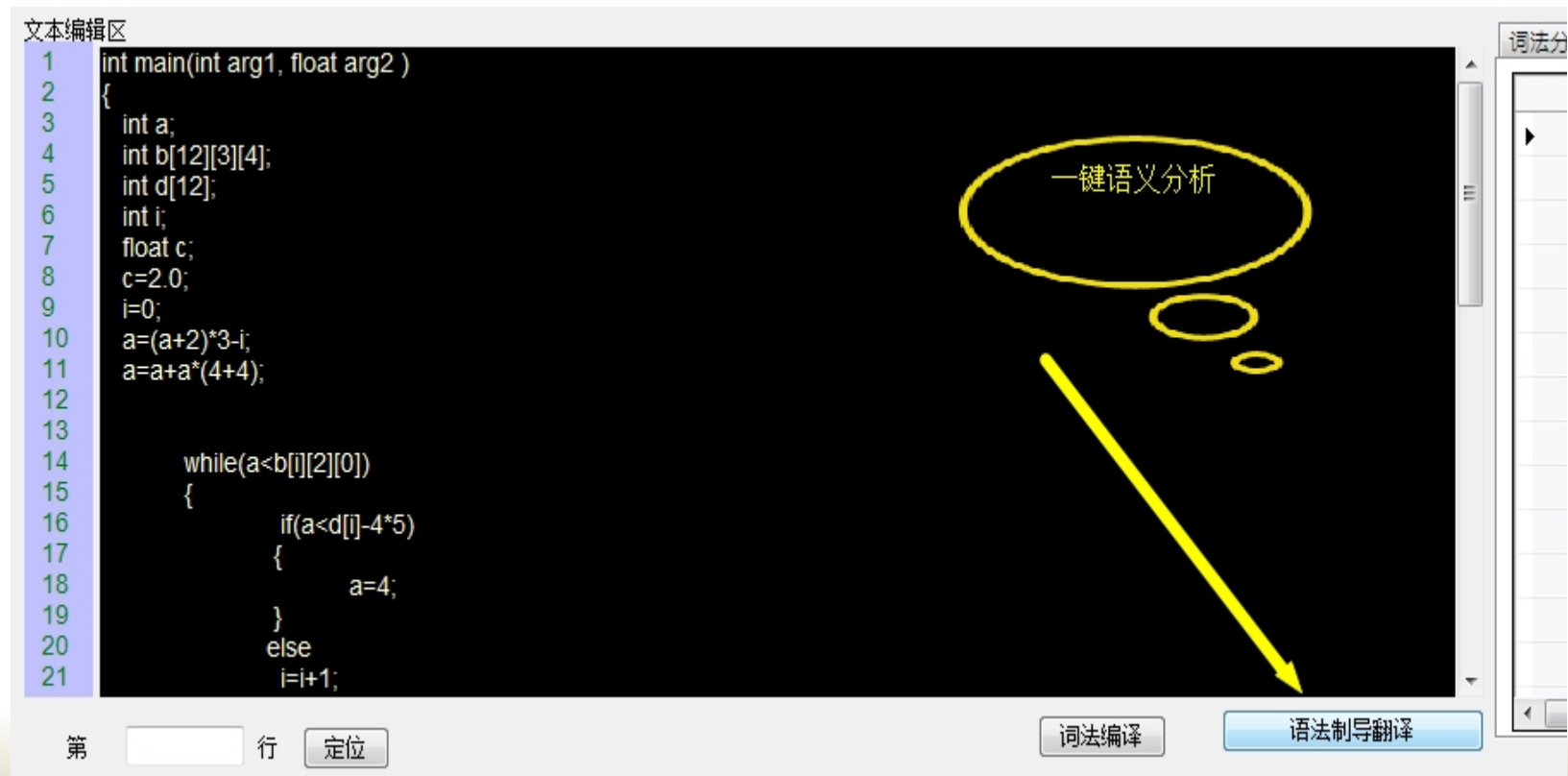
操作流程

1113730116 陈晨

步骤1 打开文法文件，点击语法分析按钮



步骤2 输入C语言文件， 点击按钮



步骤3.1 结果分析

符号表

	返回值 类型	函数名	偏移量
	int	fun	0
▶	int	main	644

点击一个函数右侧出现该函数符

类型	变量名称	所属类型	变量长度	内存地址
参数	arg1	int	4	0
参数	arg2	float	4	4
变量	a	int	4	8
变量	b	int [12] [3...]	576	12
变量	d	int [12]	48	588
变量	i	int	4	636
变量	c	float	4	640

步骤3.2 三地址码

词法分析结果 语义分析结果

三地址码

▶	c=2.0
	i=0
	t1=a+2
	t2=t1*3
	t3=t2-i
	a=t3
	t4=4+4
	t5=a*t4
	t6=a+t5
	a=t6
	L1:
	t7=48*i
	t8=16*2

点击查看三地址码

步骤3.3 错误列表

```
28 {
29   int b;
30   int a;
31   int a;
32   int b;
33   float c;
34   int d[12];
35
36   c=2.3;
37   a=c;
38   a=(a-12)*B+6;
39   d[12]=a;
40   a[11]=c;
41   d[c]=12;
42
43
44
45
46 }
47
48
```

第 行

错误列表

错误信息	错误单词	错误行
▶ 同一个函数内变量不能重复声明	a	32
同一个函数内变量不能重复声明	b	32

点击一个错误可快速定位到错误附近