

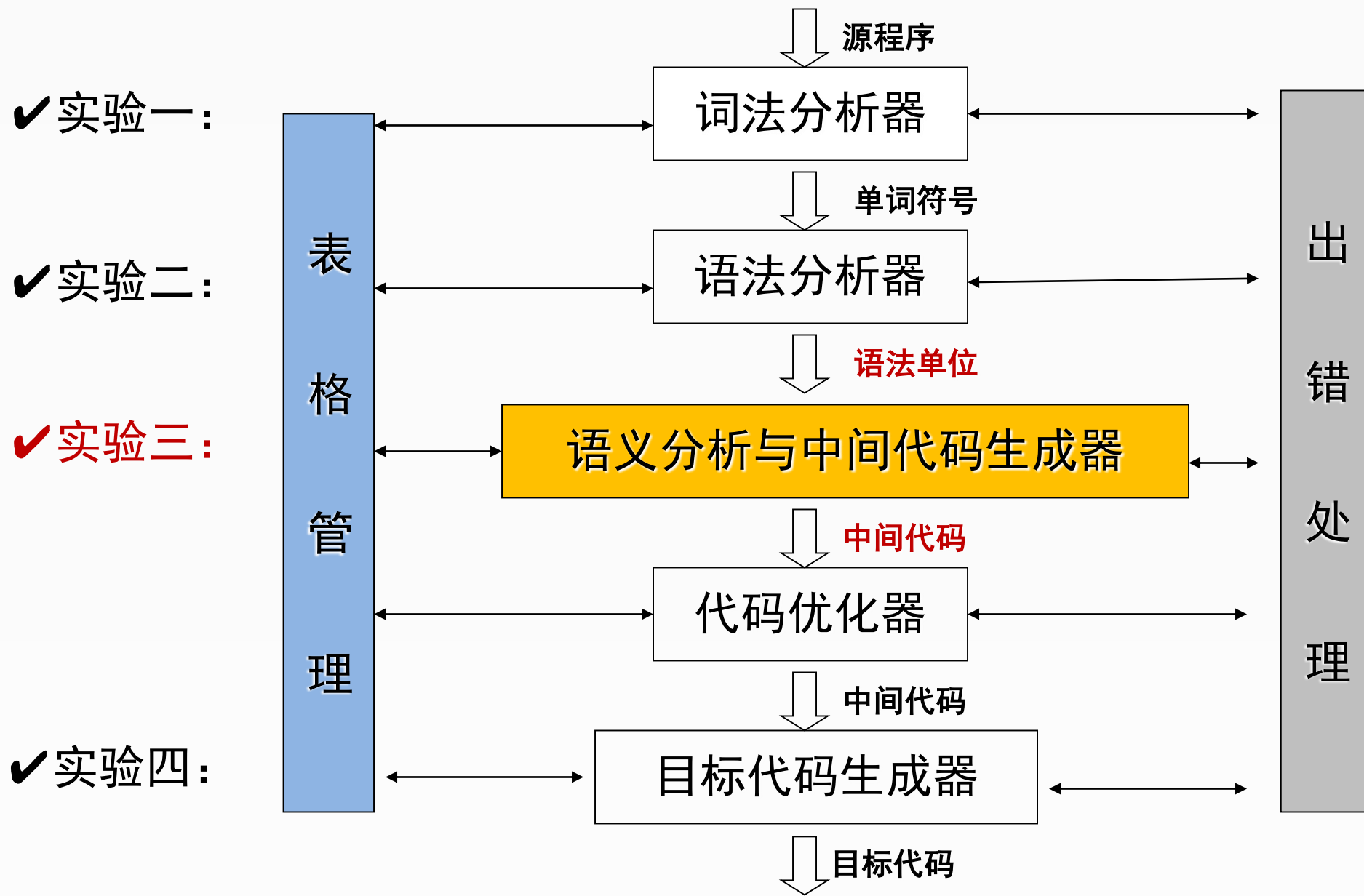


编译原理

实验三：典型语句的语义分析及中间代码生成

规格严格，功夫到家

编译程序的总体结构



实验目的

1. 加深对自顶向下语法制导翻译技术的理解与掌握。
2. 加深对自底下上语法制导翻译技术的理解与掌握。
3. 巩固对语义分析的基本功能和原理的认识，理解中间代码生产的作用。

实验学时数：4学时

实验内容

1. 针对**自顶向下**分析法或者**自底向上**分析法（二选一）中所使用的文法，在完成实验二（语法分析）的基础上为语法正确的单词串设计翻译方案，**完成语法制导翻译**。
2. 利用该翻译方案，对所给程序段进行分析，输出生成**中间代码序列**和**符号表**，并保存在相应文件中。
3. 中间代码可选三地址码的**三元式表示**或者**四元式表示**（不限）。
4. **较低完成要求**：常见**赋值语句**的语义分析与中间代码生产（书本P267）。
5. **较优完成要求**：除赋值语句外，实现带嵌套的声明语句，控制结构，布尔表达式（另选其一）的语义分析与中间代码生产。

实验总体步骤

1. 在完成实验二（语法分析）的基础上为语法正确的单词串设计**翻译方案**；
2. 利用该翻译方案，对所给程序段进行分析，**编写主函数与语义子程序**；
3. **输出**：三地址码表示的**中间代码序列**，并以**文件格式**保存；
扩展的符号表，并以**文件格式**保存；
4. 完成实验报告语义分析部分内容。

概念回顾

1. **语法制导翻译**：在进行语法分析的同时完成相应的语义处理；
2. **语法制导定义**：每一个产生式编写一个语义子程序，当一个产生式获得匹配时，就调用相应的语义子程序来实现语义检查与翻译；
3. **语法制导翻译模式**：在产生式的右部的**适当位置**，插入相应的语义动作，按照分析的进程，执行遇到的语义动作；

概念回顾

1. **S-属性定义**：只含有**综合属性**的语法制导定义称为S-属性定义；

S-属性定义在自底向上的语法分析过程中容易被实现；

2. **L-属性定义**：一个语法制导定义被称为L-属性定义，当且仅当它的每个属性或者是**综合属性**，或者是满足如下条件的**继承属性**：设有产生式 $A \rightarrow X_1X_2 \cdots X_n$ ，其右部符号 X_i ($1 \leq i \leq n$) 的继承属性只依赖于下列属性：

- (1) A的继承属性；
- (2) 产生式中 X_i 左边的符号 X_1 、 X_2 、 \cdots 、 X_{i-1} 的综合属性或继承属性；
- (3) X_i 本身综合属性或继承属性，但前提是 X_i 的属性不能在依赖图中形成回路。

L-属性定义又称为L-属性文法；

简单赋值语句的翻译

$S \rightarrow D \text{ ID}$ { $p = \text{lookup}(\text{ID.name})$; if $p == \text{nil}$ then error;
ID.type=D.type}

$D \rightarrow \text{INT}$ {D.type ==INT.name;}

Lookup (name): 查询符号表,
返回name对应的记录

$S \rightarrow \text{ID} = E$ { $p = \text{lookup}(\text{ID.name})$; if $p == \text{nil}$ then error;
gencode($p = 'E.addr$);}

gencode (code): 生成
三地址指令code

$E \rightarrow E + A$ {E.addr = newtemp(); gencode(E.addr='E.addr'+A.addr);}

newtemp(): 生成一个新的临
时变量t, 返回t的地址

$E \rightarrow E - A$ {E.addr = newtemp(); gencode(E.addr='E.addr'-A.addr);}

$E \rightarrow A * B$ {E.addr = newtemp(); gencode(E.addr='A.addr'*B.addr);}

$B \rightarrow (E)$ {B.addr = E.addr ;}

$B \rightarrow \text{ID}$ {B.addr = $\text{lookup}(\text{ID.name})$; if B.addr == nil then error;}

例: $\text{result} = a * b + (a - b) + c$;

三地址码:

$t_1 = a * b$

$t_2 = a - b$

$t_3 = t_1 + t_2$

$t_4 = t_3 + c$

$\text{result} = t_4$

学习视频: M00C 哈尔滨工业大学《编译原理》, 第12讲中间代码生成_2, 6-3 简单复制语句的翻译

<https://www.icourse163.org/learn/HIT-1002123007?tid=1450215473#/learn/content?type=detail&id=1214538629&sm=1>

输出（中间代码的三地址码的两种表示）

	op	arg ₁	arg ₂	result
0	minus	b		t ₁
1	+	c	d	t ₂
2	*	t ₁	t ₂	t ₃
3	+	c	d	t ₄
4	-	t ₃	t ₄	t ₅
5	assign	t ₅		a
	...			

图(a) 三地址码的四元式表示

三元式表示：区别只是arg1和arg2可以是某个三元式的编号，（图7.2(b)中用圆括号括起来的数字），表示用该三元式的**运算结果作为运算对象**。

	op	arg ₁	arg ₂
0	minus	b	
1	+	c	d
2	*	(0)	(1)
3	+	c	d
4	-	(2)	(3)
5	assign	a	(4)
	...		

图(b) 三地址码的三元式表示

扩展符号表（知识回顾）

符号表：以名字为关键字来记录其信息的**数据结构**；

支持的基本操作包括**插入**、**查找**和**删除**。

符号表的数据结构类型：线性表（优点：简单直观，缺点：插入时间复杂度高）

散列表（查找、插入效率高）

实验一的基础上扩展符号表

	名字	符号种类	类型	地址	扩展属性
符号表表项1	abc	变量	int	0	NULL
符号表表项2	i	变量	int	4	NULL
符号表表项3	myarray	数组	int	8	
			

维数

各维维长

2

3

4

图：多种符号共用符号表的一种实现结构

附加功能

非必要完成项，如完成，请在实验报告中另行标注实现方法，酌情加分但不超出实验部分的总分。

补全嵌套的声明语句，控制结构，布尔表达式的语义分析与中间代码生成。

实验报告内容(二选一)

1. 自顶向下语法制导翻译

- ✓ 写出适合自顶向下分析的翻译方案；
- ✓ 写出主要产生式语义翻译时的中间代码数据结构；
- ✓ 写出语义分析后生成的中间代码序列和符号表；
- ✓ 实验中用到的特色方法或设计技巧。

实验报告内容(二选一)

1. 自底向上语法制导翻译

- ✓ 写出适合自底向上分析的翻译方案；
- ✓ 设计存放中间代码的数据结构；
- ✓ 写出语义分析后生成的中间代码序列和符号表；
- ✓ 实验中用到的特色方法或设计技巧。

实验思考

实验中用到的特色方法或设计技巧，是否可以优化数据结构；
实验中遇到的问题及解决方案。

同学们，
独立开始实验