

杂项

选择

词法分析器的输出结果是**单词的种别编码和自身值**

正规式等价是指**识别的语言集相等**

四元式之间的联系是通过**临时变量**实现的

删除多余运算属于局部优化

采用 DAG 图不可以实现的优化有**删除归纳变量**

判断

所有编译程序都有目标代码生成阶段 **正确**

每个过程的活动记录的体积在编译时可静态确定 **正确**

采用三元式实现三地址代码时，不利于对中间代码进行优化 **正确**

总结

第一二章

编译：将**高级语言（源语言）** 翻译成**汇编语言或机器语言（目标语言）** 的过程

编译过程分为：**词法分析、语法分析、语义分析及中间代码生成、优化、目标代码生成**五个阶段

词法分析器：从**源程序**中识别出各个单词，确定**单词的类型**，转换为统一的**机内表示**——词法单元

语法分析器：从词法分析器输出的 token 序列中**识别出各类短语**，并构造**语法分析树**

语义分析：收集标识符的属性信息

前端：**分析部分**，与**源语言**相关；后端：**综合部分**，与**目标语言**相关

第三四十五章

自顶向下的推导总是选择每个句型的最左**非终结符**替换

存在共同前缀会导致**回溯**现象，是一种**不确定的分析**

预测分析不需要回溯，是一种**确定的**自顶向下分析方法

1. 递归的方式：在递归下降分析中，根据**预测分析表**进行产生式的选择
2. 非递归的方式：根据预测分析表构造一个**自动机**，也叫**表驱动的分析**

预测分析程序是使用一张**分析表**和一个**符号栈**进行联合控制的

自底向上分析器采用**移进、归约、接受、出错**四种操作

第六章

语句翻译的主要任务：

- 声明语句：收集标识符的**类型**等属性信息，并为每一个名字分配一个**相对地址**
- 赋值语句：生成对表达式求值的**三地址码**

类型等价：

- 结构等价（**T1** 和 **T2** 满足以下条件之一）：
 - **T1** 和 **T2** 是相同的基本类型
 - **T1** 和 **T2** 是将同一类型构造符应用于结构等价的类型上形成的
 - **T1** 是 **T2** 的类型别名
- 名字等价：两个名字完全相同

数组的内情向量表：将数组属性信息存入符号表时**扩展属性指针**指向的表，用于存放数组的各类信息

转跳代码中，逻辑运算符被翻译成**转跳指令**，而运算符本身**不出现在代码中**。

SDT 的通用实现：先建立一棵语法分析树，然后按照**从左到右的深度优先**顺序来执行这些动作

回填的基本思想：生成转跳指令时先不指定其**目标标号**。把预计目标标号相同的转跳指令放入一个**列表**中，等到能确定正确的目标标号时再去填充这些指令的目标标号，例如：

- **B.true**list 包含一个转跳指令列表，这些指令最终的目标标号就是 **B** 为**真**时要转跳到的目标指令标号
- **B.false**list 同理，**B** 为**假**时控制流将转跳的目标标号就是列表中转跳指令的最终目标标号

语义分析的错误检测类型有：

- 变量、过程**未经声明就使用**
- 变量、过程**重复声明**
- **运算分量**类型不匹配

- **操作符与操作数**之间类型不匹配

第七章

静态存储分配：在**编译时刻**就能**确定大小**的数据对象，可以在编译时刻就为它们分配存储空间

动态存储分配：不能在编译时刻完全确定数据对象的大小，就要在编译时刻仅产生各种必要信息，在**运行时刻**再**动态地分配**数据对象存储空间

静态和**动态**分别对应**编译时刻**和**运行时刻**。

活动记录：使用**过程**（函数、方法）为动作单元的语言，编译器通常以过程为单位分配存储空间，过程的每次执行称为该过程的一个**活动**，此时分配的一块连续的存储区称为**活动记录**。

静态存储分配：编译阶段为每个过程确定**活动记录**在目标程序的位置，过程的**每次**执行，名字都绑定到**相同的**存储单元。

静态存储分配的限制：

- 数组上下界是**常数**
- 不允许**递归**
- 不允许**动态建立**数据实体

顺序分配法：按照过程出现的先后顺序逐段分配存储空间，各过程活动记录存储空间互不相交

层次分配法：分析调用过程，对于无相互调用的并列过程，尽量让其局部数据共享存储空间

栈式存储分配：运行时存储以栈的形式进行管理。过程**调用**时活动记录**压入**栈、结束时**弹出**栈。这种安排使得活跃时段不相交的过程**共享空间**，并且非局部变量的**相对地址是固定的**。

活动树：描述程序**运行期间****控制**进入、离开各个活动情况的树。每个节点对应一个活动，根节点是 `main` 过程的活动。

调用序列：

- **调用者**计算实际参数位置
- **调用者**将**返回地址**放到被调用者转台字段，修改 `top-sp`
- **被调用者**保存**寄存器值**和**其他状态信息**
- **被调用者**初始化局部数据并开始执行

返回序列：

- **被调用者**将返回值放到与参数相邻的位置
- **被调用者**恢复寄存器信息和 `top-sp` 并转跳

- **调用者**通过 `top-sp` 和相对位置信息获取返回值

变长数据的存储分配：**编译时刻不能确定大小的对象**一般放在**堆区**，但是如果其是过程局部对象，也可以放在**运行时刻栈**中，这样做可以**避免**过程结束时对其进行**垃圾回收**，**减少开销**。只有数据对象在**局部过程**有效，结束时**不可访问**，才可以用**栈**来分配空间

非局部数据：一个过程除了可以使用**过程自身的局部数据**，还可以使用**过程外声明的非局部数据**

静态作用域规则：若过程 `b` 嵌套在 `a` 的声明中，则 `b` 可以访问 `a` 中声明的变量

访问链：在嵌套的活动记录之间建立的一种**指针**，使得**内嵌**过程可以访问**外层**声明的对象

Display 表：一个指针数组 `d`，任何时候 `d[i]` 都指向第 `i` 层的最新活动记录，可以解决访问链访问外层过程名字效率较低的问题

形式参数：在**过程定义**中使用的参数

实际参数：在**调用过程**时使用的参数

常用的参数传递方式有**传值、传地址、传值结果、传名**

符号表：用于存放**标识符的属性信息**的数据结构

符号表上的主要操作：

- 翻译**声明**语句：插入、查询
- 翻译**可执行**语句：查询

第八章

每个基本块由一组**总是一起执行**的指令组成

优化的分类：机器无关优化、机器相关优化、局部代码优化、全局代码优化

常量合并：在**编译时刻**推导出一个表达式的值是常量，就可以使用该常量来替代这个表达式

强度削弱：用较快的操作代替较慢的操作

数据流分析：一组用来获取**有关数据如何沿着程序执行路径流动**的相关信息的技术

在每一种数据流分析应用中，都会把每个**程序点**和一个**数据流值**关联起来

数据流分析包括：到达-定值分析、活跃变量分析、可用表达式分析

不包含其它循环的循环称为**最内循环**

到达定值：循环不变计算的检测、检测“变量未经定值就被引用”、常量传播

活跃变量：删除无用赋值、寄存器分配

可用表达式：消除全局公共子表达式、复制传播

循环不变语句移动条件

1. s 所在的基本块是循环所有出口结点的支配结点
2. 循环中没有其它语句对 x 赋值
3. 循环中对 x 的引用仅由 s 到达

循环的优化方法：循环不变表达式外提、归纳变量删除、强度削减

第九章

代码生成器的主要任务

1. 指令选择：选择适当的**目标机指令**来实现中间表示（IR）语句
2. 寄存器分配和指派：把哪个值放在哪个寄存器中
3. 指令排序：按照什么顺序来安排指令的执行

寄存器描述符：记录每个寄存器当前存放的是哪些变量的值

地址描述符：记录运行时每个名字的当前值存放在哪个或哪些位置

窥孔：程序上的一个小的滑动窗口

窥孔优化：在优化的时候，检查目标指令的一个滑动窗口

1. 冗余指令删除
2. 控制流优化
3. 代数优化
4. 机器特有指令的使用