

# 编译器和编程语言设计导论

*Introduction to Compilers and Language Design*

左元 译

2022

# 目录

|          |                    |          |
|----------|--------------------|----------|
| <b>1</b> | <b>简介</b>          | <b>1</b> |
| 1.1      | 什么是编译器？            | 1        |
| 1.2      | 为什么要学习编译器？         | 1        |
| 1.3      | 学习编译器的最佳实践是什么？     | 1        |
| 1.4      | 应该使用什么语言实现编译器？     | 1        |
| 1.5      | 这本书和其它编译器课本的区别是什么？ | 1        |
| 1.6      | 我需要阅读哪些其他编译器课本？    | 1        |
| <b>2</b> | <b>快速指南</b>        | <b>2</b> |
| 2.1      | 编译器工具链             | 2        |
| 2.2      | C 编译器的各个阶段         | 2        |
| 2.3      | 编译举例               | 2        |
| 2.4      | 练习                 | 2        |
| <b>3</b> | <b>词法分析</b>        | <b>3</b> |
| 3.1      | 标记的类型              | 3        |
| 3.2      | 一个手工词法分析器          | 3        |
| 3.3      | 正则表达式              | 3        |
| 3.4      | 有限自动机              | 3        |
| 3.4.1    | 确定性有限自动机           | 3        |
| 3.4.2    | 非确定性有限自动机          | 3        |
| 3.5      | 转换算法               | 3        |
| 3.5.1    | 将 RE 转换成 NFA       | 3        |
| 3.5.2    | 将 NFA 转换成 DFA      | 3        |
| 3.5.3    | 最小化 DFA            | 3        |
| 3.6      | 有限自动机的局限性          | 3        |
| 3.7      | 词法分析器生成器的使用        | 3        |
| 3.8      | 实践上的考虑             | 3        |
| 3.9      | 练习                 | 3        |
| 3.10     | 深入阅读               | 3        |
| <b>4</b> | <b>语法分析</b>        | <b>4</b> |
| 4.1      | 概述                 | 4        |
| 4.2      | 上下无关文法             | 4        |
| 4.2.1    | 文法推导               | 4        |
| 4.2.2    | 有歧义的文法             | 4        |

|          |                     |          |
|----------|---------------------|----------|
| 4.3      | LL 语法               | 4        |
| 4.3.1    | 消除左递归               | 4        |
| 4.3.2    | 消除最左公共前缀            | 4        |
| 4.3.3    | First 集合和 Follow 集合 | 4        |
| 4.3.4    | 递归下降语法分析            | 4        |
| 4.3.5    | 表驱动语法分析             | 4        |
| 4.4      | LR 语法               | 4        |
| 4.4.1    | 移进-归约语法分析           | 4        |
| 4.4.2    | LR(0) 自动化           | 4        |
| 4.4.3    | SLR 语法分析            | 4        |
| 4.4.4    | LR(1) 语法分析          | 4        |
| 4.4.5    | LALR 语法分析           | 4        |
| 4.5      | 语法分类重探              | 4        |
| 4.6      | 乔姆斯基文法等级体系          | 4        |
| 4.7      | 练习                  | 4        |
| 4.8      | 深入阅读                | 4        |
| <b>5</b> | <b>实践中的语法分析</b>     | <b>5</b> |
| 5.1      | Bison 语法分析器生成器      | 5        |
| 5.2      | 表达式校验器              | 5        |
| 5.3      | 表达式解释器              | 5        |
| 5.4      | 表达式树                | 5        |
| 5.5      | 练习                  | 5        |
| 5.6      | 深入阅读                | 5        |
| <b>6</b> | <b>抽象语法树</b>        | <b>6</b> |
| 6.1      | 概览                  | 6        |
| 6.2      | 声明                  | 6        |
| 6.3      | 语句                  | 6        |
| 6.4      | 表达式                 | 6        |
| 6.5      | 类型                  | 6        |
| 6.6      | 将以上内容结合在一起          | 6        |
| 6.7      | 构建 AST              | 6        |
| 6.8      | 练习                  | 6        |
| <b>7</b> | <b>语义分析</b>         | <b>7</b> |
| 7.1      | 类型系统概述              | 7        |
| 7.2      | 类型系统的设计             | 7        |
| 7.3      | B-Minor 的类型系统       | 7        |

|           |  |           |
|-----------|--|-----------|
| 7.4       | 符号表 . . . . .                              | 7         |
| 7.5       | 名字的解析 . . . . .                            | 7         |
| 7.6       | 类型检查的实现 . . . . .                          | 7         |
| 7.7       | 错误信息 . . . . .                             | 7         |
| 7.8       | 练习 . . . . .                               | 7         |
| 7.9       | 深入阅读 . . . . .                             | 7         |
| <b>8</b>  | <b>中间表示</b>                                | <b>8</b>  |
| 8.1       | 简介 . . . . .                               | 8         |
| 8.2       | 抽象语法树 . . . . .                            | 8         |
| 8.3       | 有向无环图 . . . . .                            | 8         |
| 8.4       | 控制流图 . . . . .                             | 8         |
| 8.5       | 静态单赋值形式 . . . . .                          | 8         |
| 8.6       | 线性 IR . . . . .                            | 8         |
| 8.7       | 栈机器 IR . . . . .                           | 8         |
| 8.8       | IR 举例 . . . . .                            | 8         |
| 8.8.1     | GIMPLE-GNU Simple Representation . . . . . | 8         |
| 8.8.2     | LLVM-Low Level Virtual Machine . . . . .   | 8         |
| 8.8.3     | JVM-Java Virtual Machine . . . . .         | 8         |
| 8.9       | 练习 . . . . .                               | 8         |
| 8.10      | 深入阅读 . . . . .                             | 8         |
| <b>9</b>  | <b>内存组成</b>                                | <b>9</b>  |
| 9.1       | 简介 . . . . .                               | 9         |
| 9.2       | 逻辑分区 . . . . .                             | 9         |
| 9.3       | 堆的管理 . . . . .                             | 9         |
| 9.4       | 栈的管理 . . . . .                             | 9         |
| 9.4.1     | 栈调用约定 . . . . .                            | 9         |
| 9.4.2     | 寄存器调用约定 . . . . .                          | 9         |
| 9.5       | 数据的定位 . . . . .                            | 9         |
| 9.6       | 程序的加载 . . . . .                            | 9         |
| 9.7       | 深入阅读 . . . . .                             | 9         |
| <b>10</b> | <b>汇编语言</b>                                | <b>10</b> |
| 10.1      | 简介 . . . . .                               | 10        |
| 10.2      | 开源汇编工具 . . . . .                           | 10        |
| 10.3      | X86 汇编语言 . . . . .                         | 10        |
| 10.3.1    | 寄存器和数据类型 . . . . .                         | 10        |
| 10.3.2    | 寻址模式 . . . . .                             | 10        |

|           |             |           |
|-----------|-------------|-----------|
| 10.3.3    | 基本算术        | 10        |
| 10.3.4    | 比较和跳转       | 10        |
| 10.3.5    | 栈           | 10        |
| 10.3.6    | 函数调用        | 10        |
| 10.3.7    | 叶子函数的定义     | 10        |
| 10.3.8    | 复杂函数的定义     | 10        |
| 10.4      | ARM 汇编语言    | 10        |
| 10.4.1    | 寄存器和数据类型    | 10        |
| 10.4.2    | 寻址模式        | 10        |
| 10.4.3    | 基本算术        | 10        |
| 10.4.4    | 比较和分支       | 10        |
| 10.4.5    | 栈           | 10        |
| 10.4.6    | 函数调用        | 10        |
| 10.4.7    | 叶子函数的定义     | 10        |
| 10.4.8    | 复杂函数的定义     | 10        |
| 10.4.9    | 64 位的不同之处   | 10        |
| 10.5      | 深入阅读        | 10        |
| <b>11</b> | <b>代码生成</b> | <b>11</b> |
| 11.1      | 简介          | 11        |
| 11.2      | 函数的代码生成     | 11        |
| 11.3      | 表达式的代码生成    | 11        |
| 11.4      | 语句的代码生成     | 11        |
| 11.5      | 条件表达式的代码生成  | 11        |
| 11.6      | 声明的代码生成     | 11        |
| 11.7      | 练习          | 11        |
| <b>12</b> | <b>优化</b>   | <b>12</b> |
| 12.1      | 概览          | 12        |
| 12.2      | 优化的思路       | 12        |
| 12.3      | 高层优化        | 12        |
| 12.3.1    | 常量折叠        | 12        |
| 12.3.2    | 强度削减        | 12        |
| 12.3.3    | 循环展开        | 12        |
| 12.3.4    | 代码提升        | 12        |
| 12.3.5    | 函数内联        | 12        |
| 12.3.6    | 死代码检测和删除    | 12        |
| 12.4      | 底层优化        | 12        |
| 12.4.1    | 窥孔优化        | 12        |

|                            |    |
|----------------------------|----|
| 12.4.2 指令选择 . . . . .      | 12 |
| 12.5 寄存器分配 . . . . .       | 12 |
| 12.5.1 寄存器分配的安全性 . . . . . | 12 |
| 12.5.2 寄存器分配的优先级 . . . . . | 12 |
| 12.5.3 变量之间的冲突 . . . . .   | 12 |
| 12.5.4 全局寄存器分配 . . . . .   | 12 |
| 12.6 优化的陷阱 . . . . .       | 12 |
| 12.7 优化的相互影响 . . . . .     | 12 |
| 12.8 练习 . . . . .          | 12 |
| 12.9 深入阅读 . . . . .        | 12 |

# 第一章 简介

1.1 什么是编译器？

1.2 为什么要学习编译器？

1.3 学习编译器的最佳实践是什么？

1.4 应该使用什么语言实现编译器？

1.5 这本书和其它编译器课本的区别是什么？

1.6 我需要阅读哪些其他编译器课本？

## 第二章 快速指南

### 2.1 编译器工具链

### 2.2 C 编译器的各个阶段

### 2.3 编译举例

### 2.4 练习



## 第三章 词法分析

### 3.1 标记的类型

### 3.2 一个手工词法分析器

### 3.3 正则表达式

### 3.4 有限自动机

#### 3.4.1 确定性有限自动机

#### 3.4.2 非确定性有限自动机

### 3.5 转换算法

#### 3.5.1 将 RE 转换成 NFA

#### 3.5.2 将 NFA 转换成 DFA

#### 3.5.3 最小化 DFA

### 3.6 有限自动机的局限性

### 3.7 词法分析器生成器的使用

### 3.8 实践上的考虑

### 3.9 练习

### 3.10 深入阅读

## 第四章 语法分析

### 4.1 概述

### 4.2 上下无关文法

#### 4.2.1 文法推导

#### 4.2.2 有歧义的文法

### 4.3 LL 语法

#### 4.3.1 消除左递归

#### 4.3.2 消除最左公共前缀

#### 4.3.3 First 集合和 Follow 集合

#### 4.3.4 递归下降语法分析

#### 4.3.5 表驱动语法分析

### 4.4 LR 语法

#### 4.4.1 移进-归约语法分析

#### 4.4.2 LR(0) 自动化

#### 4.4.3 SLR 语法分析

#### 4.4.4 LR(1) 语法分析

#### 4.4.5 LALR 语法分析

### 4.5 语法分类重探

### 4.6 乔姆斯基文法等级体系

### 4.7 练习

### 4.8 深入阅读

## 第五章 实践中的语法分析

### 5.1 Bison 语法分析器生成器

### 5.2 表达式校验器

### 5.3 表达式解释器

### 5.4 表达式树

### 5.5 练习

### 5.6 深入阅读

## 第六章 抽象语法树

### 6.1 概览

### 6.2 声明

### 6.3 语句

### 6.4 表达式

### 6.5 类型

### 6.6 将以上内容结合在一起

### 6.7 构建 AST

### 6.8 练习

## 第七章 语义分析

### 7.1 类型系统概述

### 7.2 类型系统的设计

### 7.3 B-Minor 的类型系统

### 7.4 符号表

### 7.5 名字的解析

### 7.6 类型检查的实现

### 7.7 错误信息

### 7.8 练习

### 7.9 深入阅读

## 第八章 中间表示

### 8.1 简介

### 8.2 抽象语法树

### 8.3 有向无环图

### 8.4 控制流图

### 8.5 静态单赋值形式

### 8.6 线性 IR

### 8.7 栈机器 IR

### 8.8 IR 举例

#### 8.8.1 GIMPLE-GNU Simple Representation

#### 8.8.2 LLVM-Low Level Virtual Machine

#### 8.8.3 JVM-Java Virtual Machine

### 8.9 练习

### 8.10 深入阅读

## 第九章 内存组成

### 9.1 简介

### 9.2 逻辑分区

### 9.3 堆的管理

### 9.4 栈的管理

#### 9.4.1 栈调用约定

#### 9.4.2 寄存器调用约定

### 9.5 数据的定位

### 9.6 程序的加载

### 9.7 深入阅读

# 第十章 汇编语言

## 10.1 简介

## 10.2 开源汇编工具

## 10.3 X86 汇编语言

### 10.3.1 寄存器和数据类型

### 10.3.2 寻址模式

### 10.3.3 基本算术

### 10.3.4 比较和跳转

### 10.3.5 栈

### 10.3.6 函数调用

### 10.3.7 叶子函数的定义

### 10.3.8 复杂函数的定义

## 10.4 ARM 汇编语言

### 10.4.1 寄存器和数据类型

### 10.4.2 寻址模式

### 10.4.3 基本算术

### 10.4.4 比较和分支

### 10.4.5 栈

### 10.4.6 函数调用

### 10.4.7 叶子函数的定义

### 10.4.8 复杂函数的定义

### 10.4.9 64 位的不同之处

## 10.5 深入阅读



# 第十一章 代码生成

## 11.1 简介

## 11.2 函数的代码生成

## 11.3 表达式的代码生成

## 11.4 语句的代码生成

## 11.5 条件表达式的代码生成

## 11.6 声明的代码生成

## 11.7 练习

## 第十二章 优化

### 12.1 概览

### 12.2 优化的思路

### 12.3 高层优化

#### 12.3.1 常量折叠

#### 12.3.2 强度削减

#### 12.3.3 循环展开

#### 12.3.4 代码提升

#### 12.3.5 函数内联

#### 12.3.6 死代码检测和删除

### 12.4 底层优化

#### 12.4.1 窥孔优化

#### 12.4.2 指令选择

### 12.5 寄存器分配

#### 12.5.1 寄存器分配的安全性

#### 12.5.2 寄存器分配的优先级

#### 12.5.3 变量之间的冲突

#### 12.5.4 全局寄存器分配

### 12.6 优化的陷阱

### 12.7 优化的相互影响

### 12.8 练习

### 12.9 深入阅读