**课程编号：C0801003030**

**编译方法结课作业**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **姓名** | **张文祥** | **学号** | | **20175088** |
| **班级** |  | **任课教师** | | **刘洪娟** |
| **开设学期** | **2019-2020春季学期** | | | |
| **评定成绩** |  | | **评定人** | **刘洪娟** |
| **评定日期** | **2020年4月20日** |

**东北大学软件学院**

题目1：编译程序中使用的关键技术都有哪些应用方向，请详细说明。

1，有限自动机应用到正则表达式中

词法分析中的有限自动机，是正则表达式匹配的基础，通过将正则表达式解析为有限自动机，实现文本匹配，正则表达式引擎有两类：一种是以DFA（确定性有穷自动机）线性执行，不要求回溯，在lex和Mysql中使用DFA。另一种是NFA（非确定性有穷自动机）需要回溯（也称为贪婪模式），以指定顺序测试正则表达式的所有可能的扩展并接受第一个匹配项。因为传统的 NFA 构造正则表达式的特定扩展以获得成功的匹配，所以它可以捕获子表达式匹配和匹配的反向引用，在Java和Python中使用NFA

对于正则表达式为/ya(msen|nsen|nsem)/ ，在DFA引擎中，会构造出来三个确定性有限自动机分别是 yamsen 和 yansen 和 yansem 依次使用这三个自动机对文本进行匹配，而NFA中则只构造一个非确定性有限自动即可，因为在非确定性有限自动机中可以进行回溯，只需一个。

2，结构编辑器（例如eclipse）

结构编辑器将一个命令序列作为输入来构造一个源程序。结构编辑器不仅实现普通文本编辑器的文本创建和修改功能，而且还对程序文本进行分析，为源程序构造恰当的层次结构，结构编辑器能够完成程序准备过程中所需要的功能，例如检查输入格式是否正确，自动提供关键字（当用户输入while时，编辑器能自动提供匹配的关键字do并提醒用户必须在二者之间插入一个条件体），能够从左括号跳转到右括号，这类结构编辑器，就类似于一个编译器语法语义分析阶段的输出。

3，智能打印机

智能打印机能够对程序进行分析，打印出结构清晰的程序。例如，注释以一种特殊的字体打印，根据各个语句在程序的层次结构中的嵌套深度来缩排这些语句。

4，文本格式器

文本格式器输入字符流，输入字符流中的多数字符串是需要排版输出的字符串，同时字符流中也包含一些用来说明字符流中的段落，图表或者上下标的数学结构的命令。这些是词法分析阶段的输出。

题目2：窥孔优化的概念？窥孔优化的常用技术有哪些？

1，窥孔优化的概念

窥孔优化，是一种很局部的优化方式，编译器仅仅在一个基本块或者多个基本块中，针对已经生成的代码，结合CPU自己指令的特点，通过一些认为可能带来性能提升的转换规则，或者通过整体的分析，通过指令转换，提升代码性能。别看这些代码转换很局部，很小，但可能会带来很大的性能提升。

这个窥孔，可以认为是一个滑动窗口，编译器在实施窥孔优化时，就仅仅分析这个窗口内的指令。每次转换之后，可能还会暴露相邻窗口之间的某些优化机会，所以可以多次调用窥孔优化，尽可能提升性能。

2，窥孔优化常用的技术

窥孔优化可以在四个方面寻找优化机会：冗余指令删除，;控制流优化；强度削弱；利用特有指令。

冗余指令删除：（1）删除死代码：有些代码可能用于不会被执行到，这样在窥孔优化阶段，如果发现这样的代码，就可以直接删除。典型的方式是优化双跳转，即第一条跳转指令的目的地址还是一条跳转指令时，可以删除后一条跳转指令，并修改第一条跳转指令的目标地址。另外，对于不可能进入的分支也可以使用这种方式删除。（2）删除冗余load和store，但是这种指令序列的转换和合成有个前提，必须保证这些指令按照顺序执行，即这些指令之间，不能有其他标号，即入口。也就是说这些指令必须在一个基本块中。当然，你也可以在编译器较前面阶段的优化中，针对该操作，做变换。这样到了窥孔优化时，就不再会有这样的代码了。

控制流优化：中间代码生成阶段，很可能经常产生一些跳转到跳转指令，跳转到分支跳转、分支跳转到跳转之类的指令，都可以在窥孔优化中想办法解决掉，当然你也可以在中间代码中优化它们。一些分支被删除后，可能还存在一些不会被到达的标号(label)，也可以顺便删除之，这样就会提升基本块的大小，增加优化机会。

强度削弱：即利用代价较小的指令或操作替代代价较大的指令或操作，从而提升性能。比如x=x+0, x=x\*1之类的操作就能直接避免，x=x\*2,x=x/2之类的操作可以使用左移或右移实现。x^2之类的指数运算可以削弱为x\*x的乘法运算，浮点数除以常数的运算可以转换为浮点数乘以常数的倒数。这些都是优化方式。

充分利用特有指令：CPU都会提供一些特殊指令完成特殊操作，比如DSP芯片中可能有复杂的数字信号处理指令，龙芯中有乘加指令以及一些向量扩展指令。还有一些CPU可能提供自增、自减、取绝对值指令。这些都能在窥孔优化中生成。提升程序的运行性能。

**成绩评价表格**：

|  |  |
| --- | --- |
| 考核标准 | 得分 |
| （1）正确理解和掌握编译程序的工作原理；（30%）； |  |
| （2）有自己独到的见解（20%）； |  |
| （3）文字描述能够使用专业术语完成（30%）； |  |
| （4）论述有理论依据（20%） |  |