FG2015

----ACM compiler 2015 project

高宇

pmkfb@aliyun.com

**摘要**

这篇文档中, 我介绍FG2015, 我的ACM compiler 2015 project. 同时也包括了我的实现方法, 其中有得到满分的方法和最速得到基本分的方法.

**关键字**

C语言, 编译器, 自编译

# 简介

FG2015(未来道具2015), 是一个能够自编译的C语言编译器, 支持C语言的一个子集. 显然它是用C语言写的. 它实现了多种窥孔优化和linear scan寄存器分配以及二次寄存器分配. 它拥有解释器功能, 能够模拟支持的语言中无键盘输入的所有程序. 它还能打印single static form, 但是没有利用single static form 做任何优化.

# 代码框架

下图展示了FG2015的代码框架, 即代码中各个功能模块的交互方式. 这个代码框架能被精简从而用于编写一个编程复杂度最低(\*)的ACM compiler2015 project.

中间代码

中间代码

获取词

带优化的语法分析器

词法分析器

优化器

调用

解释器

带语义检查的中间代码生成器

# 词法分析器

## nextToken()

词法分析器唯一的对外借口是nextToken()函数, 它返回程序的下一个token. 下面介绍每种token的处理方法.

## 空白和注释

nextToken()函数开头必须跳过当前所有的空白和注释. 空白是指空格和制表符以及换行符, 注释则是单行注释和多行注释.

## 关键字和标识符

当跳过空白和注释后, 遇到的第一个字符是字母(‘\_’, ‘$’, ‘A’..’Z’, ‘a’..’z’)时, 获取后面连续的字母和数字作为一个字符串. 如果在关键字表里匹配不到, 就是标识符. 关键字可以以字符串或编号的方式存储在token中, 标识符则必须包含字符串.

## 整数常量

当跳过空白和注释后, 遇到的第一个字符是数字, 获取后面连续的字母和数字作为一个字符串, 并解析为8进制/16进制/10进制字符串常量, 将其值存储在token中.

## 字符常量和字符串常量

当跳过空白和注释后, 遇到的第一个字符是单引号或双引号时, 就获取后面连续的(转义)字符, 直到同样的引号出现.

SSA输出

## 衔接符

衔接符(‘\’)必须在词法分析前处理, 衔接符能吃掉它后面紧随的换行. 但是换行必须考虑操作系统的区别.

代码生成器

# 语法分析器

## 概述

由于支持的语言子集使用CFG定义的, 所以语法分析器采用了最简单暴力的递归下降+看2个token.\

## 支持的语言

### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized and 6-points of white space above the subsubsection head.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

#### Subsubsections

The heading for subsubsections should be in Times New Roman 11-point italic with initial letters capitalized.

# ACKNOWLEDGMENTS

Our thanks to ACM SIGCHI for allowing us to modify templates they had developed.

# REFERENCES

1. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI= <http://doi.acm.org/10.1145/161468.16147>.
2. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.
3. Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 526-531. DOI= <http://doi.acm.org/10.1145/332040.332491>.
4. Tavel, P. 2007. *Modeling and Simulation Design*. AK Peters Ltd., Natick, MA.
5. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
6. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 3 (Mar. 2003), 1289-1305.
7. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada, November 02 - 05, 2003). UIST '03. ACM, New York, NY, 1-10. DOI= <http://doi.acm.org/10.1145/964696.964697>.
8. Yu, Y. T. and Lau, M. F. 2006. A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions. *J. Syst. Softw.* 79, 5 (May. 2006), 577-590. DOI= <http://dx.doi.org/10.1016/j.jss.2005.05.030>.
9. Spector, A. Z. 1989. Achieving application requirements. In *Distributed Systems*, S. Mullender, Ed. ACM Press Frontier Series. ACM, New York, NY, 19-33. DOI= <http://doi.acm.org/10.1145/90417.90738>.

Columns on Last Page Should Be Made As Close As Possible to Equal Length