北京大学

本科生毕业论文

题目： 中文题目 title 楷体，二号，加粗

英文题目（Times New Roman，三号，加粗）

姓　　名：杨子毅

学　　号：1600011063

院　　系：信息科学技术学院

本科专业：软件工程

指导老师：胡振江

二〇二〇　年　五　月

北京大学本科毕业论文导师评阅表

<table>
<tr><td>学生姓名</td><td></td><td>学生学号</td><td></td><td>论文成绩</td><td></td></tr>
<tr><td>学院 (系)</td><td colspan="3"></td><td>学生所在专业</td><td></td></tr>
<tr><td>导师姓名</td><td></td><td>导师单位/
所在研究所</td><td></td><td>导师职称</td><td></td></tr>
<tr><td colspan="2">论文题目
（中、英文）</td><td colspan="4"></td></tr>
<tr><td colspan="6">导师评语
（包含对论文的性质、难度、分量、综合训练等是否符合培养目标的目的等评价）

导师签名：

年　　月　　日</td></tr>
</table>

版权声明

摘要

随着计算机科学的普及和编程语言的发展，编程语言、特别是领域特定语言的应用越来越日常化。语法糖作为实现领域特定语言的一项重要技术在近年来发展火热，相关的研究（引用）以及为 DSL 而诞生的编程语言（引用）都在进展显著。我们对一个重组糖（语法糖的一项工作）系列的文章进行学习，并基于 PLT Redex 设计出一个轻量级重组糖算法，简单实现了一套工具并在一些例子上进行测试。结果显示我们的轻量级重组糖算法具有（xxx）等优势

关键词: 领域特定语言、语法糖、函数式语言、解释器

Abstract

Key Words:

全文目录

第一章 章节名称

1. 一级段落名称

1.1 二级段落名称

1.1.1 三级段落名称

引用如[1]，引用表如表 1，引用图如图 1.

表 1 不同频率下的输入和输出阻抗

频率 (Hz)	1	10k	1M
输入电阻 (Ω/°)	339.719k/-87.84	5.6707k/-9.827	351.188/-72.377
输出电阻 (Ω/°)	338.638k/-89.663	1.9866k/-1.1228	1.9189k/-14.801

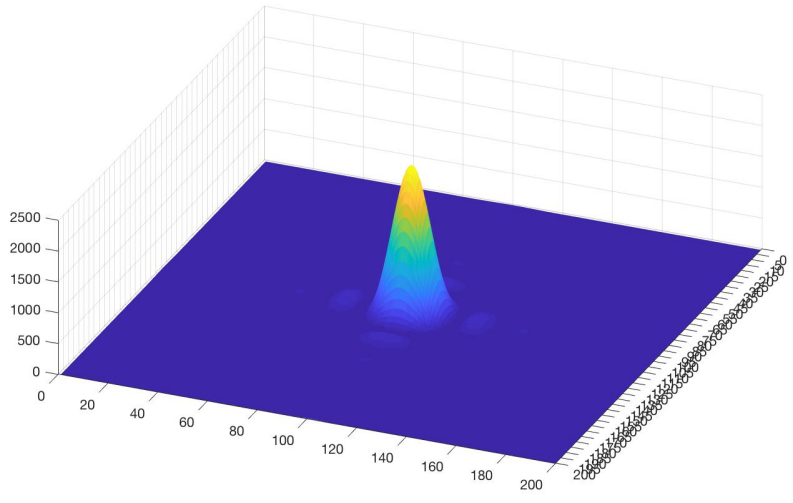

图 1 示例图片

第二章 绪论

1. 研究背景

领域特定语言的研究及应用日益广泛，其中不乏正则表达式、SQL、XML等应用场合及其多的领域特定语言。而随着计算机科学技术的发展，更多新颖的的场景（引用）加入到了使用 DSL 的队伍。而在一般场合中，语言设计者是计算机科学家，语言使用者却是领域内的专家。因此，当领域特定语言的使用出现一些问题时，领域内的专家（使用者）需要得到领域特定的信息来进行处理。

语法糖是近些年一种流行的领域特定语言实现方法，其方法源于 Lisp 的宏系统（Macro），经过 Scheme 语言的发展，再到 Racket 语言的扩展。（可以引用 From …那篇的部分）其优点是：。。。

然而，语法糖的一项缺陷，导致其应用领域一直局限于计算机科学内部。我们先来看下面这个例子。

（举一个开药方/饭店流程的例子）

将这个例子抽象到简单的例子 and(or(1,0),and(1,0))

我们可以看到，在执行过程中，语法糖被展开成 if，在通用语言中继续执行，得到最终结果。但实际应用到特定领域时，我们很明显不希望得到这样复杂的执行过程，特别是对于对计算机内部语言不熟悉的领域专家来说，这种执行过程是没有意义的。

2. 解决的问题及研究现状

我们将上述问题总结为语法糖解糖的单向性，对上面的例子的 xx 行，我们可以看出，其存在等价的领域特定语言表示 and(1,and(1,0))。如果语法糖的解糖有一个逆过程（重组糖），我们就可以找到其对应的这样一个序列。我们将在第二章对问题进行形式化定义。我们在后文，将领域特定语言视为外部语言，通用语言视为内部语言。

3. 相关工作及存在的问题

我们的方法主要借鉴和对比 Resugaring 系列一些工作，其中前两篇和本文的工作紧密相关，我们将简单讲述一下这两篇工作的方法及缺陷

第一篇介绍的方法，基本思想是将内部语言的求值序列每一步加上标签，进行搜索，试图得到其对应的在外部语言的表示。

其算法希望具有如下三个性质：

仿真性/抽象性/覆盖性（没有被证明）

第二篇工作在第一篇的基础上，新增了两个优点：1. 解决卫生宏 2. 拓展语法糖规则（超过 syntax-rule）3. 覆盖性得到形式化证明

然而该工作仍然存在一些问题：

1. 语法糖规则依然不够丰富

2. 算法定义繁琐，通用性差。

4. 基本思路及全文结构

目前只有基本思路

本文基于语义工程工具 PLT Redex，设计了一个全新的语法糖——重组糖框架。其想法源于完全 β 规约的完备性，如下图。

（图）

该例子中，对一个结构化（称为 S 表达式，定义在第二章）的语言，定义其基础语义和规约规则。在没有限制其上下文规则（求值顺序）的前提下，生成了其规约的流程图。我们可以看出，在该图中，既包含了将语法糖展开的规约，也包含了我们期待的 and() 等等这些中间求值路径。因此我们希望使用基于规约语义的 PLT Redex，实现一个轻量级的重组糖，在这个完全图中提取出我们需要的重组糖序列。

在实现过程中，我们发现生成中国完全图并不是必须的，而我们可以在从最初的表达式开始**每次进行单步规约，在一条或多条规约规则中选择我们需要的那条规约规则** (是本工作的核心算法)，且该规则的多次执行保证重组糖的三个基本重要性质。则在这个核心算法迭代执行过程中，会留下一个对应的求值序列，在其中提取出符合输出规则的中间序列，则此序列即为重组糖的输出。

5. 本文主要贡献

1. 我们针对现有重组糖的算法进行改进，得到新的轻量级重组糖算法。

2. 我们的算法相对于现有重组糖算法，支持更多语法糖特性。

3. 我们的轻量级重组糖算法在对领域特定语言的解释器进行自动生成的研究提供了一个思路。

第三章 背景知识

1. 重组糖形式化定义

对于给定求值规则的内部语言 CoreLang，和在 CoreLang 基础上用语法糖构造的 SurfLang；对于任意 SurfLang 的表达式，得到其在 SurfLang 上的求值序列且该求值序列满足三个性质：

1. 仿真性：求值序列需要和在 CoreLang 上的求值顺序相同，即存在 CoreLang 上的求值序列中的部分中间过程与该序列中的元素对应。该性质是重组糖有意义的前提。

2. 抽象性：求值序列中只存在 SurfLang 中存在的术语，没有引入 CoreLang 中的术语。该性质是重组糖研究的目的。

3. 覆盖性：在求值规则中没有跳过一些中间过程。该性质不是正确性的必要条件，却是在应用中极其重要的；加上前两条性质满足的正确性，构成了重组糖的全部重要性质。

2. 完全 β 规约、归约语义和 PLT Redex

与 β 规约的概念不同，完全 β 规约是一种基于 β 规约的求值顺序规则。对于一个嵌套的 lambda 表达式，每个表达式都可能进行 β 规约，而常规的 call-by-name 和 call-by-value 都是对规约顺序进行了约定，而完全 β 规约就是一种不定序的求值规则，每个可 β 规约的位置都有可能进行规约，因此得到的规约路径不是一条，而是一个图，且这个图的起点和终点只有一个。如图所示的例子就是一个完全 β 规约的求值图

（图）

可以看出，在完全 beta 规则中，对任何位置的可 beta 规约的 lambda 表达式，都可以进行规约。因此，与 call-by-name 和 call-by-value 不同的是，这种求值规则是不定序的。

本文工作的最初思想就是基于完全 beta 规约。

规约语义：我们需要在求值规则中约定每一个表达式的规约规则，。。。PLT Redex 是基于此语义的语义工程工具。

第四章 算法形式化定义

1. 对语言的规定

对于给定求值规则的内部语言 CoreLang，和在 CoreLang 基础上用语法糖构造的 SurfLang；对于任意 SurfLang 的表达式，得到其在 SurfLang 上的求值序列且该求值序列满足三个性质：

首先，我们需要将语言限定在基于树形表达式的结构化语言。

树形表达式：此处我们使用类似 Lisp 的 S 表达式的递归树，基础定义如下。

Exp::= (Headid Exp*)

| Value

| Variable

结构化：对于每个表达式中的子表达式，其规约规则只和子表达式本身有关。

此外：我们对 CoreLang 和 SurfLang 进行一些简单的限制。

对 CoreLang，任意一个 Exp（每个子表达式都不包含 surfexp）最多只能有一条规约路径。这一点约束并不过分，为了保证每个程序只能有一条执行路径。

对 SurfLang，任意一个语法糖只能有一个 CoreLang 的表达式与之对应。

在 PLT Redex 中，我们将 CoreLang 和 SurfLang 视为同一个语言。则当我们定义了一个语言内部各种规约规则后，对于任意 Exp 都有其对应的一条或多条规约规则。根据对 CoreLang 的约定，有多条规约规则的表达式必然存在 SurfLang 的表达式。

为了区分 CoreLang 的语言和 SurfLang 的语言，我们将表达式的文法定义为如下

Exp::= Coreexp

| Surfexp

| Commonexp

| Otherexp

Coreexp::= (CoreHead Exp*)

Surfexp::= (SurfHead (Surfexp|Commonexp)*)

Commonexp ::= (CommonHead (Surfexp|Commonexp)*)

| value | variable

OtherSurfexp ::= (SurfHead Exp* Coreexp Exp*)

OtherCommonexp ::= (CommonHead Exp* Coreexp Exp*)

可以看出，在我们的重组糖方法中，可以输出的表达式是 Surfexp 和 Commonexp，即不存在任何子表达式中存在 Coreexp。

2. 算法描述

核心算法 f 定义如下：

对于每个表达式 Exp，我们将对它所有规约规则中选择一条符合 resugaring 的仿真性规则的规约，且尽可能不破坏任何语法糖。

1. 如果 Exp 是 Coreexp 或 Commonexp 或 OtherCommonexp，则其规约规则或是将表达式规约到另一个表达式，此时只有一条规则；

2. 或是其规约不满足导致内部子表达式需要规约，此时因为 CoreLang 的定序性，只会有一个子表达式被规约（且此表达式为 Surfexp），此时对该子表达式 Subexp 递归调用核心算法 f 得到 Subexp’，则将此 Exp 中子表达式 Subexp 规约为 Subexp’的规则就是我们需要的规则。

3. 如果 Exp 是 Surfexp 或 OtherSurfexp，如果内部子表达式无可规约的，则必然会展开该语法糖；

4. 如果存在可规约的子表达式对于每个子表达式，如果可规约，则根据我们的设定，存在一条关于此子表达式的规约规则。因此每个子表达式都可能被规约的前提下，我们需要对 Surfexp 或 OtherSurfexp 的语法糖进行展开为 Exp’（此展开只有一种规约规则对应），之后对 Exp’调用核心算法 f，检测内部哪个子表达式在 f 调用过程中被规约，则此子表达式需要在 Exp 处首先被规约，对应对该子表达式进行规约的规约规则即为所需要的路径。

参考文献

[1] 期刊

[2] 网络文档

1. 期刊作者. 论文名. 刊名, 出版年份, 卷号 (期号): 起始-截止页

2. 专著作者. 书名. 版本 (第一版不写). 出版城市: 出版社, 出版年份: 起始-截止页

3. 论文集论文作者. 论文题目//编者. 论文集名: 其他题名信息. 出版城市 (或者会议城市): 出版者, 出版年: 引文起始-截止页码

4. 学位论文作者. 学位论文题名. 城市: 论文保存单位, 年份

5. 网络文献作者. 题名 [文献类型标志/文献载体标志]. 出版地: 出版者, 出版年 (更新日期)[引用日期]. 获取和访问路径

* 注意: 作者姓前名后, 超过 3 名作者列前 3 名, 后加“, 等”; 英文姓名, 姓前名后, 姓首字母大写, 名缩写; 文献的项目要完整, 各项的顺序和标点要和格式要求一致; 未公开发表的论文、报告不列入正式文献, 如有必要可在正文当页下加注。英文文献格式同上。参考文献在正文中按出现顺序用 [1], [2]..... 在右上角标注, 放在“参考文献”中时, 用 [1], [2], ... 顺序标注。

参考文献

[1] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935.

这是参考“参考文献”，主要用来看引用的顺序，请手动些参考文献或自行写程序，最终编译请删除

本科期间的主要工作和成果

本科期间参加的主要科研项目
本研基金

1. 基金名称. 基金类型. 指导老师. 基金支持年限

各种科研项目

1. 项目名称. 项目类型

格式下
期刊：
全部作者. 论文名. 期刊名, 出版年份, 卷号 (期号): 起始-截止页
会议论文：
全部作者. 论文名. 会议名, 会议举办地, 会议举办时间, 起始-截止页
专利
全部专利申请人. 专利名称. 专利申请号. 专利申请日期. 国别

致谢