

SHANGHAI JIAO TONG UNIVERSITY

本科生毕业设计（论文）开题报告



论文题目： 基于数值型输入输出样例的程序合成器

学生姓名: 王绍宇

学生学号: 5080379072

专 业: 软件工程

指导教师: 赵建军

学院 (系): 软件学院

教务处制表

**填表说明**

1. 根据《上海交通大学关于本科生毕业设计(论文)工作的若干规定》要求，每位学生必须认真撰写《毕业设计（论文）开题报告》。
2. 每位学生应在指导教师的指导下认真、实事求是地填写各项内容。文字表达要明确、严谨，语句通顺，条理清晰。外来语要同时用原文和中文表达，第一次出现的缩写词，须注出全称。
3. 开题前，须进行文献查阅，要求与论文研究有关的主要参考文献阅读数量不少于10篇，其中外文资料应占一定比例。参考文献的书写请参照《上海交通大学本科生毕业设计（论文）撰写规范》。
4. 毕业设计（论文）开题报告总字数应满足本院（系）要求。
5. 请用宋体小四号字体填写，并用A4纸打印，于左侧装订成册。
6. 该表填写完毕后，须请指导教师审核，并签署意见。
7. 《上海交通大学本科生毕业设计（论文）开题报告》将作为答辩资格审查的主要材料之一。
8. 本表格不够可自行扩页。

**毕业设计(论文)开题报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 论文题目 | 基于数值型输入输出样例的程序合成器 | | | | | |
| 课题来源 |  | 课题性质 | 设计 | 项目编号 |  | |
| 课题研究目的和意义（含国内外研究现状综述）：  面对日益增长的数据信息，人们越来越渴望能使用自动化的方法，将数据中有用的信息提炼成易于展示和理解的形式。成千上万的终端用户每天面对着海量的数据，进行着重复而乏味的整理工作，耗费了大量的劳力。采用自动化的方法，如程序合成，可以使这些终端用户从乏味的重复劳动中解放出来，利用计算机强大的计算优势，生成用户想要的处理程序来处理他们的工作。  程序合成，原来的相关工作一般被归为自动编程。该概念并不是很新了，不过较早的研究都受计算机的计算能力、搜索等方面的技术成熟度等的限制，最终成果并不显著。近期，该领域又重新被人们重视，微软的一个研究组在研究中取得了一定的成果，并初步总结了相关问题的方法论。不过，这个领域现在的发展还处于初级阶段，相关的研究还有很长的路要走。  程序合成的任务就是由某些约束形式的用户意图来生成可执行程序。它做的事情与编译器不同，编译器是将输入程序写成有结构性的语言，通常执行表达式导向的翻译；而合成器能够接受多样的混合形式的约束（比如，输入输出样例，示范，输入输出间的逻辑关系，自然语言，程序片段或低效的程序），通常执行某些程序的空间下的某种搜索。[1]简言之，给定高级的约束说明和低级的编程语言之后，我们的目标就是自动地合成符合约束条件的有效的程序。[3]  我将根据已有的程序合成的基本方法，结合相关领域的较成熟的各项技术，如人工智能中的自然语言的识别、机器学习、专家系统、知识库等等，[2]搜索技术中，如穷举搜索，版本空间代数，逻辑推理技术等等，完成一个简单的程序合成器。我的程序合成器将能够根据用户给定的数列的部分项，推测出其中的规律，生成足以表达完整数列的程序。并通过用户的反馈进行调整，或者进行重新生成。  程序合成，很有潜力开启编程语言的一个新纪元，把我们从比较低级的how（怎么做）语言带入到更高级的what（是什么）语言的时代。[1]我所进行的研究，应用前景也是十分广泛的。对于一维的数值规律发现，可以应用于教育领域，如IQ测试的解答数组找规律的问题等等；作为引擎，可以为编程人员提供快速生成代码的接口；数据中的模式发现，在对数据进行分析预测中发挥着相当重要的作用；如果扩展到二维，很可能应用到对图像和声音等多媒体信息的理解，缺失信息的修复，信息压缩等等领域中。 | | | | | | |
| 课题研究内容：  程序合成中主要有这样三个维度[1]：  约束的种类：约束即用户意图的表达，这可以是输入与输出间的逻辑关系，输入输出样例，示范，自然语言，程序片段或低效的程序等等。  程序的搜索空间：可以是命令式或函数式程序（需要缩减控制结构，限制操作符集合），上下文无关语法/正则表达式，简洁的逻辑表达式等等。  所需的搜索技术：可以基于穷举搜索，版本空间代数，机器学习技术，逻辑推理技术等等。  在“研究方法和研究思路”中我将阐明本项目关于第二三维度的相关问题。下面是关于第一个维度，约束的种类所做的说明。  我决定将输入输出样例作为用户意图的表达形式。  我把程序运行的过程分为三个元素，即程序的输入，程序的算法，程序的输出。通常，程序本身就是算法的体现。把实现好的“算法”（程序或函数）以及“输入”交给计算机，计算机都给出确定的“输出”[[1]](#footnote-1)。而如果将这种流程变换一下，给定输入和输出，让计算机生成由输入到输出的内在联系（算法）就成了我要研究的问题。  不过这个问题具有相当的难度。因为，输入，算法，输出，这三个元素并非给定任何两者都能推出第三者。比如一些不可逆的加密算法，已知了算法本身，从输出并不能有效地转化回输入。而从输入和输出推出算法，难度又更进一步。  于是，对于初步研究，为了化简问题，我决定在以下方面对我的项目做一些限制：  首先，我只针对数值型输入输出的函数进行合成。计算机可执行的程序，一般都可以抽象成为函数，而函数的输入和输出有着多种数据类型。并且有些函数除了返回值之外会产生一些附加效果。这些情况都过于复杂，于是，我只针对其中最简单的情况，对无附加效果，单一数值作为输入和输出的函数进行合成。并且我将把主要精力放在整数的处理上，因为对于离散的计算机世界，整数已经具有足够的能力处理全部逻辑和运算问题。  然后，对于函数的逻辑的复杂度进行限制。目前，限于搜索的代价，程序合成技术只能生成那些逻辑简单的小段程序。所以，我只选取对数值进行有限步简单地运算那样的程序进行合成，预计生成的程序代码不过只有十几行到几十行。  最后，如果需要，还会结合用户给出的反馈进行合成。正如同许多哲学问题那样，程序合成没有标准答案。合成的结果如何，还要用户来评判。所以，结合用户的反馈（如给出反例，在备选答案中做出选择等）来进行合成，能使结果更加符合用户意图。 | | | | | | |

|  |
| --- |
| 研究方法和研究思路（技术路线）：  上文中提到，程序合成中的三大维度。并简单解释了我关于第一个维度——“约束的种类”所采用的方法。下面说明我对第二三维度的解决方案的初步思路。  对于“程序的搜索空间”，我应设计出一种形式化语言来表示对数值的操作处理。通过对语言的约束来减小搜索的空间范围。  这就要解答“什么样的语言是更好的”的问题。[6]  首先，程序体短小的比冗长的要好。我们将采用抽象语法树的大小来评判程序体的大小。  然后，我们通常认为非递归比递归要好，如果非要递归，单递归比多重递归要好。  此外，函数只应包含少量的特殊约束。比如，递归的终止条件，最好只有列表或树为空时，或者0,1等这样“自然的”终止条件。  对于“搜索的技术，”我决定采用机器学习的技术，现在还在初步探究中。我还要研究一下相关的SAT/SMT solver （可满足问题求解器）——如微软的Z3等。在我的研究中搜索这一块可能会有一定帮助。 |

|  |
| --- |
| 预期研究结果：（可选填）  1. 一个基于数值型输入输出样例的程序合成器。  2. 一篇学位论文。  3. 开发过程中的相关技术文档。  4. 可能会在会议或期刊上发表相关论文。 |

|  |
| --- |
| 计划进度安排：  2012.1 - 2012.2 背景研究、技术学习、资料积累  2012.2 - 2012.3 学习Z3，开发领域相关语言  2012.3 - 2012.4 系统设计实现与调试  2012.4 - 2012.5 系统参数配置调优  2012.5 - 2012.6 系统进一步完善，采集实验数据，撰写论文  2012.6 课题答辩 |

|  |
| --- |
| 参考文献：  [1] Sumit Gulwani. Dimensions in program synthesis. In ACM Symposium on PPDP, 2010.  [2] D E Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Publisher: Addison-Wesley, 1989.  [3] Shachar Itzhaky, Sumit Gulwani and Neil Immerman. A Simple Inductive Synthesis Methodology and its Applications. In Proceedings of the ACM international conference, on OOPSLA 2010.  [4] Saurabh Srivastava, Sumit Gulwani, Swarat Chaudhuri and Jeffrey S. Foster. Path-based Inductive Synthesis for Program Inversion. In Proceedings of the 32nd ACM SIGPLAN conference, on PLDI 2011.  [5] Shachar Itzhaky, Sumit Gulwani, Neil Immerman and Mooly Sagiv. A Simple Inductive Synthesis Methodology and its Applications. In Proceedings of the ACM international conference on OOPSLA 2010  [6] Pieter Koopman and Rinus Plasmeijer. Synthesis of Functions Using Generic Programming. Lecture Notes in Computer Science, 2010, Volume 5812/2010, 25-49, DOI: 10.1007/978-3-642-11931-6\_2 |

|  |
| --- |
| 指导教师意见（课题难度是否适中、工作量是否饱满、进度安排是否合理、工作条件是否具备、是否同意开题等）：  指导教师签名：  年 月 日 |
| 学院（系）意见：    审查结果： □ 同 意 □ 不 同 意  学院（系）负责人签名：  年 月 日 |

1. 随机算法不在讨论范围内。对于伪随机算法，如果将种子也算作输入，那输出也是可以说是确定的。 [↑](#footnote-ref-1)