

# Design and Implementation of Automatic Marking System for Programming Questions Based on Script Technique

Yali Liu<sup>1</sup>, Wenyan Chai<sup>1</sup>, Xiurong Li<sup>2</sup>, Xiaoxing Chu<sup>3</sup>

<sup>1</sup>) School of Science and Technology, Beijing City University, Beijing, China (liuyali@bcu.edu.cn, cwycwyhd1@bcu.edu.cn)

<sup>2</sup>) College of Computer Science and Technology, Beijing University of Technology, Beijing, China (xiurong\_li@bjut.edu.cn)

<sup>3</sup>) Company of Na\_software Technology Co., Ltd., Beijing, China (xiaoxing.chu@gmail.com)

**Abstract**—After analyzing current status of automatic marking methods for programming questions, a novel script-based approach was proposed for designing marking rule. This paper studied the design and implementation of an automatic marking system. The system marks console programs written in a plurality of programming languages. It is composed of two parts, a dynamic-test module and a static-test module based on the script-based approach. Code-marking rules are written in script files. In the dynamic-test module, the programs of the students are compiled, executed and marked with test cases. In the static-test module, the script files are analyzed and executed with both the programs and the outputs of the dynamic-test module. Suggestions would be given to improve the quality of those console programs. Experiments showed that the static-test module enhances functions of the marking system.

**Keywords**—automatic marking, code-marking rule, script, programming questions

## 基于脚本技术的编程题自动测评系统设计与实现

刘雅丽<sup>1</sup> 柴文妍<sup>1</sup> 李秀荣<sup>2</sup> 储小星<sup>3</sup>

<sup>1</sup>) 北京城市学院信息学部, 北京, 中国

<sup>2</sup>) 北京工业大学计算机学院, 北京, 中国

<sup>3</sup>) 北京纳信科技有限公司, 北京, 中国

**摘要** 分析了编程题自动测评方法的现状, 提出了一种基于脚本技术的源代码测评方法, 设计并实现了一种使用脚本来设计代码评分规则的自动测评系统。系统由动态测试模块和基于脚本的静态测试模块组成, 能测评多种语言的控制台程序。动态测试模块编译、运行并输入测试用例, 计算运行得分。静态测试模块通过自定义评分规则, 以源代码、运行时间等信息为输入计算代码评分得分, 并对被测评者的源代码提出改进意见。实验表明基于脚本技术的测评方法增强了测评系统的功能。

**关键词** 自动测评, 代码评分规则, 脚本, 编程题

### 1. 引言

随着教育信息化的进展, 计算机语言程序设计及其相关课程的作业提交方式已经转为提交电子源代码, 考试方式也转变为上机考试。但是对教师来说, 对学生源代码给出及时、正确、一致的评价, 却是非常沉重的负担。不仅步骤繁琐, 而且手工测评也难以客观地评价学生源程序。

本文提出了基于脚本技术的编程题自动测评系统 NJudge, 是一种基于 B/S 结构的 Windows 操作系统环境下的测评系统, 能识别并测评多种语言的控制台应用程序。

### 2. 编程题自动测评方法现状

#### 2.1 编程题自动测评方法现状

从被测评对象的角度分类, 目前对控制台程序编程题的自动测评方法主要有四种。

- (1) 采用动态测试的方法测评源代码的正确性: 仅仅针对可执行文件的执行结果进行测评[1-3]。
- (2) 采用静态测试的方法测评源代码的正确性[4-7]。
- (3) 采用静态测试的方法测评源代码质量[8]。
- (4) 结合动态测试和静态测试, 测评源代码的正确性[9-10]。

\*北京市财政项目支持 (资助号: PXM2012\_014202\_000201)

## 2.2 自动化测试中的脚本技术

测试脚本是测试自动化中必要组成部分[11]。文献设计了场景—事件驱动脚本语言 SED\_TSL [12]。文献提出了一种面向嵌入式软件测试的脚本语言 CATScript[13]。在编程题自动测评系统中引入脚本技术，可以很大程度上提高测评系统的灵活性，且便于测评系统进行功能扩展。

本文提出的 NJudge 测评系统选择了 NScript 脚本语言作为静态测试的脚本语言[14]。采用脚本技术可以精确地分析源代码。在 NScript 脚本中不仅可以灵活制定源代码评分规则，而且还可以对动态测试中获得的信息（如编译结果、执行时间、占用内存等）进行评分以及分析，进而可以对被测评者的源代码提出改进意见。

## 3. NJudge 自动测评系统的设计与实现技术

### 3.1 NJudge 自动测评系统的设计

系统支持学生在线提交作业、在线考试、即时测评、在线查询成绩，支持教师在线发布练习、在线测评、代理测评、在线查询成绩、在线统一公布测评成绩等。系统总体架构图如图 1 所示。图 2 为系统自动评分模型。

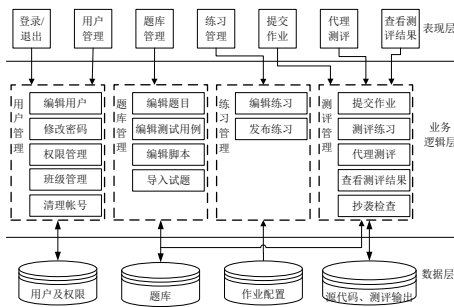


图 1 系统总体架构图

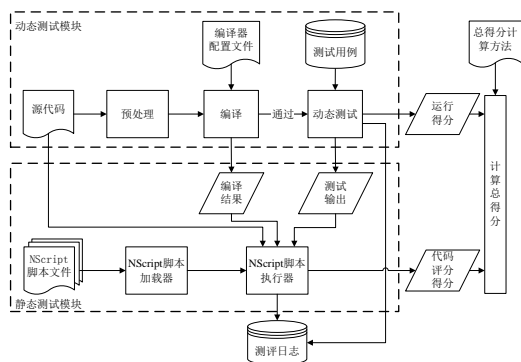


图 2 系统自动评分模型

在动态测试模块中，将该题目的若干个测试用例的输出数据与被测评程序执行结果相匹配，并根据匹配结果计

算运行得分。在静态测试模块中，NScript 脚本加载器将包含代码评分规则的脚本文件编译为汇编命令集；NScript 脚本执行器执行汇编命令集。在执行过程中，根据源代码、编译结果和测试输出，计算代码评分得分，并对被测评者的源代码提出改进意见。得到运行得分和代码评分得分后，依据教师为该题目设置的总得分计算方法，将两个分数值求和或者取最大值，即为源代码的测评得分。

系统内嵌 C、C++、Java 编译器。若要支持其它语言的自动测评，只需要在系统编译器配置文件 Compiler.cfg 中增加新语言编译器的相应配置参数，则可以支持该语言的自动测评，但是要求该编译器必须能通过命令行编译和运行。若在文件中删除或注释某种语言配置信息，则测评系统将忽视这种语言源程序，不对其自动测评。它是 NJudge 系统自动识别程序设计语言的关键，内容如图 3 所示。图 3 中为 C、C++ 和 Java 语言的编译器配置。测评系统根据源代码文件的扩展名（扩展名）决定使用哪种编译器（Compiler）以及如何编译（编译参数）、使用哪种加载器（Loader）以及如何加载编译结果文件（Loader 参数）；并通过错误标志（错误标志）识别源代码是否编译通过。

```
//可用内容: {0}=源文件全路径, {1}=QuestionName, {2}=源文件全名, {3}=exe 全名  
//扩展名, Compiler, Loader, 错误标志, 编译参数, Loader 参数:  
C, $(ModulePath)DEV-CPP\Bin\gcc.exe, , error, "{2}" -o {1}.exe, ;  
CPP, $(ModulePath)DEV-CPP\Bin\g++.exe, , error, "{2}" -o {1}.exe, ;  
JAVA, $(ModulePath)JDK6\bin\javac.exe, $(ModulePath)JDK6\bin\java.exe, 错误, "{2}", {1};
```

图 3 编译器配置文件

在作业管理模块的作业配置中，NJudge 系统为作业的每道题目均提供三种测评方法：仅动态测试，仅静态测试以及动态测试和静态测试相结合的测评方法。将两者结合起来使得测评的结果更加贴近人工测评的结果，并且使得测评对象的范围更广。使用 NScript 脚本技术的静态测试方法的核心思想是：在脚本中将标准答案的关键字、变量符号、分支语句、循环语句、类定义等信息编写进评分规则中，自动测评系统在执行脚本之前先对源代码进行必要的词法分析，再利用 NScript 脚本语言强大的字符串处理函数来进行程序源代码的分析，主要用字符串匹配的方式评分。NScript 脚本语言还提供了分支语句、循环语句等的字符串查找匹配函数，为教师撰写评分规则脚本提供了方便。使用脚本技术，不仅可以及时反馈给学生测评分数，而且能反馈是否使用了题目要求的方法来答题、是否能够满足题目的时间和空间要求、并且能够对学生的答题方法给出改进意见。

### 3.2 NJudge 自动测评系统的实现技术

在动态测试模块中，在预处理阶段读取系统配置，读



2”的详细测评日志。表1中列出了此次作业测评时间（每项时间值均为五次测评时间的平均值）。

### 4.3 实验结果分析

NJudge 系统使得测评结果更贴近教师人工测评按步骤给的分。从图9的测评日志可知，脚本中定义的8个得分点均测评通过，代码测评得满分；该同学仅仅考虑了整数末位为2的情况，因此五个动态测试用例中仅仅通过了一个，运行得分10分，总得分60分。

试题名称	试题中文名称	时间限制 (ms)	内存限制	代码检查总分	运行总分	总分计算方法	操作
count1	统计素数个数	1000	128M	50	50	取总和	修改 删除
digitpic	输出图形	1000	128M	50	50	取总和	修改 删除
find2	找2	1000	128M	50	50	取总和	修改 删除
palind1	回文数判断	1000	128M	50	50	取总和	修改 删除
peach	桃子记	1000	128M	50	50	取总和	修改 删除
series1	数列生成	1000	128M	50	50	取总和	修改 删除
series2	三角形数	1000	128M	50	50	取总和	修改 删除
series3	找数	1000	128M	50	50	取总和	修改 删除
seven	逢七过	1000	128M	50	50	取总和	修改 删除
square1	完全平方数判断	1000	128M	50	50	取总和	修改 删除

图7 十道题目的作业测评配置

题号	学号姓名	总分	日志	count1	digitpic	find2	palind1	peach	series1	series2	series3	seven	square1
12	1211640210徐博洋	950	日志	100	100	100	100	50	100	100	100	100	100
12	1211640208刘博宇	930	日志	80	100	100	100	50	100	100	100	100	100
12	1211632116戴德宝	910	日志	100	100	100	100	50	100	100	60	100	100
12	1211632125陈华中	902	日志	100	100	100	100	100	100	50	60	100	92
12	1211640314余浩	892	日志	100	100	100	92	100	100	100	100	100	100
12	1211640209任皓	882	日志	100	100	100	92	100	100	100	100	100	100
12	1211632116陈诗婷	882	日志	100	100	100	72	50	100	100	80	100	80

图8 部分测评结果

```
(22.3)E:\Nasoft\2012\_Bin\Exercise\第五周作业\12通信\1211632123强博强\find2.c
试题中文名称: 找2
代码: #include <stdio.h>
#include <stdio.h>
int main()
{
    int n,i,s=0;
    scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        if(i%2==0)
            s++;
    }
    printf("%d",s);
    return 0;
}
编译选项: 1.in
编译选项: 2.in
编译选项: 3.in
编译选项: 4.in
完全正确: 5.in; 用例0as: 内存450,560E; 得分10.00
代码检查: (6分) #include <stdio.h>
代码检查: (7分) int main () {
代码检查: (6分) scanf gets getchar ();
代码检查: (6分) printf puts putchar ();
代码检查: (6分) for char long float
代码检查: (7分) if () 或 ? ; 或 switch () | case ;
代码检查: (6分) for char long float
代码检查: (6分) 编译选项
代码检查总分: 50; 测试数据总分: 10; 取总和后总分: 60
```

图9 某个同学的测评日志

表1 十道题目的测评时间（秒）

动态和静态测试方法	动态测试方法	静态测试方法
178.815	178.550	7.754

表1中列出83名学生共681份C语言源代码的测评时间。第一列与第二列之差为真正的静态测试时间，即脚本文件编译和执行时间，而第三列的时间中大部分是扫描目录文件的时间。由此可知，测评83个目录681份源代码，静态测试花费的真实时间为265毫秒。与动态测试相比较，静态测试时间非常短。

## 5. 结束语

NJudge 系统能自动识别并测评多种结构化程序设计和面向对象程序设计语言程序。实验结果表明这种结合动态测试和静态测试的测评方法使得测评的结果更加贴近人工测评的结果，并且基于脚本技术的静态测试使得测评对象的范围更广，增强了编程题自动测评系统的功能。

### 参考文献(References)

- [1] Z. Yu, "Auto-Scoring System for C Language Source Program," Computer Era, vol. 4, pp. 6-8, April 2010. (In Chinese)
- [2] Z. Han, G. Wang and J. Han, "Programming online Judge System based On API Automatic Test," Computer Systems & Applications, vol. 7, pp. 9-13, July 2008. (In Chinese)
- [3] B. Cheang, A. Kurnia, A. Lim, et al, "On automated grading of programming assignments in an academic institution," Computers & Education, vol. 41, no. 2, pp. 121-131, 2003.
- [4] B. Lou, "Design of auto-marking software framework for Java programming questions," Computer Engineering and Design, vol. 31, no. 24, pp. 5343-5346, 5358, 2010. (In Chinese)
- [5] S. She and S. Zhou, "Application of Regex in Auto-Checking Paper of Programs," Computer Technology and Development, vol. 17, no. 7, pp.224-246, 2007. (In Chinese)
- [6] P. Ma, T. Wang and X. Su, "Auto Grading of Student Programs Based on Program Understanding," Journal of Computer Research and Development, vol. 46, no.7, pp. 1136-1142, 2009. (In Chinese)
- [7] K. Naude, J. Greyling and D. Vogts, "Marking student programs using graph similarity," Computer & Education, vol. 54, pp. 545-561, 2010.
- [8] K. Ala-Mutka, T. Uimonen and H. Jarvinen, "Supporting Students in C++ Programming Courses with Automatic Program Style Assessment," Journal of Information Technology Education, vol. 3, no. 1, pp. 245-262, 2004.
- [9] S. Qiao, Z. Liu and B. Zhu, "Looking over the Examination Paper of Programs by Computer," Application Research of Computers, vol. 8, pp. 229-231, August 2004. (In Chinese)
- [10] X. Su, Y. Wu, T. Wang, et al, "Automatic grading system for C programming examination oriented to assessment of comprehensive practical abilities," Experimental Technology and Management, vol. 27, no. 10, pp. 174-177, October 2010. (In Chinese)
- [11] M. Fewster and D. Graham, Software Test Automation: effective use of test execution tools, Addison-Wesley Publishing Co., 1999.
- [12] G. Yu and Z. xu, "Research on Automatic Safety Tests of Train Control System for Dedicated Passenger Line Based on Script Technique," Journal of the China Railway Society, vol. 33, no. 12, pp. 56-64, December 2011. (In Chinese)
- [13] S. Xu and N. Sang, "A Script Language Oriented Embedded Software Test-CATScript," Computer Engineering, vol. 31, no. 11, pp. 82-84, June 2005. (In Chinese)
- [14] Company of Na\_software Technology Ltd, NScript Technology, <http://www.nasoftgate.com>, November 2013