

doi:10.3969/j.issn.1672-4348.2014.03.007

Java 语言程序设计题自动阅卷技术研究

郑志明¹, 郑燕娥²

(1. 湄洲湾职业技术学院 基础部, 福建 莆田 351254; 2. 仰恩大学 计算机与信息学院, 福建 泉州 362014)

摘要: 程序设计题自动阅卷技术是程序设计语言考试实现无纸化的重点与难点所在。分析了现有自动阅卷技术的特点, 提出基于 XML 的采分点匹配技术, 将标准答案引入 XML 结构、采用正则表达式匹配的方法进行评分。实验结果证明, 该系统运行稳定, 与人工评分的结果大致相符。

关键词: 自动阅卷; Java; XML

中图分类号: TP312.2

文献标志码: A

文章编号: 1672-4348(2014)03-0237-04

The research of Java programming tests automatic grading technology

Zheng Zhiming¹, Zheng Yan-e²

(1. Basic Courses Section, Meizhouwan Vocational Technology College, Putian 351254, China;

2. College of Computer and Information Science, Yang-en University, Quanzhou 362014, China)

Abstract: The key to the realization of on-line programming language tests lies in the automatic grading technology of programming. The characteristics of available automatic grading technology were analysed. The key points matching technology based on XML was proposed. The standard answer was introduced into the XML structure, and scores were given by the method of regular expression. Experimental results show that the system is stable and the results are consistent with the manual grading.

Keywords: automatic grading; Java programming test; XML

随着现代信息技术的飞速发展以及互联网的普及, 在线考试已成为各高校考核学生学习能力的主要手段, 其中计算机自动阅卷技术对于大规模的考试来说意义重大。客观题的自动阅卷技术已经发展成熟, 但主观题尤其程序设计题的阅卷技术却停滞不前。相对而言, 目前比较成熟的模型为基于程序理解的编程题自动评分模型以及基于语义相似度比较的编程题自动评分模型^[1], 这两种模型都采用中间表示形式, 然后将程序整体作为比较对象, 不仅时空成本高, 而且实现难度大, 同时也失灵活性。

本文从 Java 程序设计题的语法结构及特征出发, 结合标准答案需准确、灵活地描述代码形式

的多样化, 这就需要一种可以有效描述层次结构的数据格式来存储标准答案, XML 文档正好具有这种优势, 故本文基于 XML 的采分点匹配技术, 使用正则表达式匹配的方法进行评分。

1 设计思想

1.1 自动阅卷算法分析

如何检测程序的编程思想是自动评分的要点, 评分方式兼顾了对可执行代码的动态测试与源程序的静态分析^[2-3], 对不同的考生程序采用以下两种不同的评分方法:

(1) 对于可编译运行且输出结果正确的考生程序, 为避免考生投机取巧未按要求作答, 对有输

入要求或对象属性有赋初值的考生程序,都可利用 Java 反射机制进行黑盒测试,从标准答案 XML 文档中提取预先指定的输入值并赋值予之,然后将程序运行得到的输出值赋予变量,最后将该变量的值与 XML 文档中预定的输出值进行比较,如果一致,则说明程序功能正确,那么评分结果为满分,假若程序功能错误,那么就在满分的基础上适当扣测试分。

(2) 对于可编译运行但输出结果不正确或者无法通过编译的考生程序,则进行源程序的静态分析,使用正则表达式描述模板程序的采分点语句,然后搜索考生程序是否含有这些语句,根据匹配情况给分。自动阅卷的流程如图 1 所示。

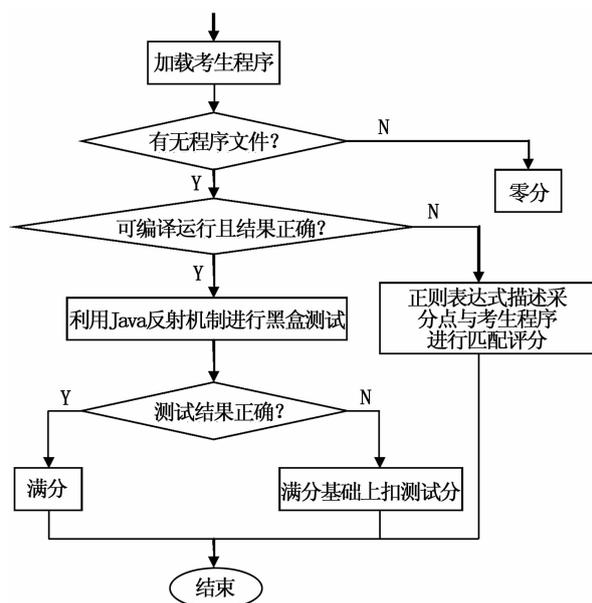


图 1 自动阅卷的流程

Fig. 1 Automatic grading process

1.2 一题多解问题的解决

程序设计题中,如果指定某种方法作答,评分相对而言比较方便,但这样限定了考生的逻辑思维,不能全面反映考生的应用能力。因此,考试过程中一般不限制解答方法。对于一题多解的问题,本文考虑将所有可能出现的解答方法,存放在 XML 结构的答案文件,用正则表达式描述所有解答方法与考生答案进行匹配给分^[4]。

1.3 采分点匹配评分技术

正则表达式是采分点匹配的核心,正则表达式构造的好坏直接决定了自动阅卷系统的准确率。正则表达式并不能分析语义,只能使用它构

造一个尽可能通用并尽量减少歧义的模式来和考生程序进行匹配评分^[5]。采分点匹配评分技术的实现过程如下所示:

1) 用正则表达式描述采分点。

正则表达式在文本匹配方面的功能强大、灵活而且高效,它可以很方便把多种不同解题方法一并描述,有效解决一题多解的问题。这种匹配评分方式也比较接近人工评阅方式,使得评分过程更加合理、准确。

比如,在考生程序中经常会涉及到的循环语句,主要有 3 种,分别为: for 语句、while 语句及 do while 语句。这 3 种循环语句用正则表达式描述如下所示:

```
for \\( ([ \\w ; * ] * ) ? ; ([ \\w ; * ] * ) ? ; ([ \\w ; * ] * ) | ; \\( | while \\( ( \\w + ) ( , \\w + \\w + ) ? < \\d + \\) \\s \\{ \\w + = \\w + \\w + \\w + ; ( ? ( 2 ) ( \\s ) | ( 1 ) ( 2 ) ; ) \\} ? | do \\{ \\w + = \\w + \\w + \\w + \\; \\w + \\w + \\w + \\} }
```

```
while \\( ( \\w + < \\d + \\) \\) ;
```

其中,“\\”是正则表达式在 java 中的写法,第一个反斜杠代表转义符,第二个反斜杠才是正则表达式中的反斜杠,随后的字符具有特殊意义。比如,\\w + 代表一个或多个字符。

2) 采分点的提取。

采分点存储于标准答案 XML 文档中,匹配评分前需从 XML 文档中提取采分点。第一步:采用 DOM 解析 XML 文档生成一个文档对象。第二步:利用 XPath 对象调用 evaluate() 来计算 XPath 表达式并返回该文档对象的一个节点或节点集的节点对象,该节点对象是所要提取的采分点在 XML 文档中对应的节点。第三步:通过该节点对象调用 getChild() 获取第 1 个子节点,即当前节点本身,再通过调用 getNodeValue() 便可获取当前节点的文本数据(采分点的值)。

3) 采分点的匹配评分。

采分点的匹配评分则需要通过 java.util.regex 包中的 Pattern 和 Matcher 这两个类来实现^[6]。首先,Pattern 类调用静态方法 compile() 将一个正则表达式的字符串编译成一个 Pattern 对象;其次,通过 Pattern 对象调用 matcher() 方法和待匹配的字符串产生一个 Matcher 对象;最后通过 Matcher 对象调用相应的方法(比如,find、group、start、end 等方法)来判断各种类型的匹配

成功与否,从而给出相应的得分。

2 系统设计

标准答案引入 XML 结构,是自动阅卷系统设计的一个亮点。标准答案中需要存放多个试题的参考答案,每个试题有不同的评分方法,一种评分方法可包含多个得分项,某个得分项又有多种表示形式,XML 文档正好适合描述这种层次结构。下面给出标准答案的 XML 文档模板(以试题为单位):

```
<standardanswer >
  <itemnumber id = " 1" fullscore = " 满分分
值" >
    <itemcontent > 试题内容 </itemcontent >
    <gradingstandard method = " 1" >
      <testcasepart testscore = " 分值" >
        < testcase-input > 输入值 </testcase-
input >
          < testcase-output > 输出值 </
testcase-output >
        </testcasepart >
      </gradingstandard >
    <gradingstandard method = " 2" >
    <keypointpart >
      <keypoint1 score = " 分值" > 第 1 个
采分点 </keypoint1 >
      ...
      <keypointN score = " 分值" > 第 N 个
采分点 </keypointN >
    </keypointpart >
  </gradingstandard >
</itemnumber >
</standardanswer >
```

其中 < itemnumber id = " 1" fullscore = " 满分分值" > id 表明题号为第一题,fullsort 表明该题满分分值。< gradingstandard method = " 1" > 表明第一种评分方法,不同的评分方法通过 method 属性来区分,< keypointpart > 标记表明采分点部分,通常一个试题包含多个采分点,分别使用子标记 keypoint1...keypointN 表示,每个采分点的值均采用正则表达式描述,可有效解决代码多样化的问题。

3 实验结果分析

为全面验证自动阅卷系统的精确率,笔者抽取 6 道具有代表性的 Java 编程题进行实验,每道题抽取 100 个考生答案进行评阅^[7]。试题内容如下:

- (1)编写一个 Java 程序,输出 1 000 之内所有的完数。
- (2)编写一个程序绘制一张笑脸。
- (3)设计一个模拟的文字编辑器,并用菜单实现退出的功能。
- (4)设计一个模拟用户从银行取款的应用程序。设某银行的帐户存款额的初值是 2000 元,用线程模拟两个用户从银行取款的情况。
- (5)建立一个文本文件,输入学生 3 门课成绩。编写一个程序,读入这个文件中的数据,输出每门课的成绩的最小值、最大值和平均值。
- (6)编写一个应用程序,实现从一个数据库的某个表中查询一个列的所有信息。

系统运行结果与人工批阅结果比较见表 1:

表 1 系统运行结果与人工批阅结果比较
Tab. 1 Comparison between automatic grading system running results and artificial grading results

题号	系统评阅			人工评阅		
	因缺少	因遗漏		因缺少	因遗漏	
	MP > 模板	MP 错误	MP	MP > 模板	MP 错误	MP
	95%	<95%	<95%	95%	<95%	<95%
	/%	的数量	的数量	/%	的数量	的数量
		/个	/个		/个	/个
1	87.5	12.0	0	94.0	0	6
2	97.0	0	6	98.0	0	0
3	94.0	0	0	89.0	0	8
4	89.0	10.0	0	94.0	0	16
5	92.2	0	6	89.6	0	0
6	96.0	0	0	90.4	0	0
平均值	92.6	3.7	2	92.5	0	5

系统评阅过程中,缺少模版的原因在于一些特殊算法没能在标准答案中给出,从而影响评分准确率,而遗漏错误的原因在于本文利用 Java 平台自带的编译器对考生程序进行编译,会出现有些语义未能检测出来,从而导致评分误差。

评分准确率(MP)的计算公式如下:

$$MP = (1 - |人工评阅 - 系统评阅| / 试题总分)$$

值) $\times 100\%$

通过对实验数据的分析表明,系统评分结果与人工评阅具有相似的均值,误差比较小,从而验证该系统具有较高的准确率。

4 结论

该系统的特色是答案文件以 XML 结构来存储,便于后期的分析和处理,对于 web 考试具有较

好的实用价值。同时,利用正则表达式描述知识要点与考生程序进行匹配评分,改善了现有的自动评分模型时空成本高以及失灵活性的缺陷,但正则表达式构造的恰当与否直接影响自动阅卷系统的精确度,为此本文利用正则表达式测试器来协同系统开发。实验结果表明,评分结果大致与人工评阅相符,基本能够客观、公正评价考生编程思想。

参考文献:

- [1] 余石泉. 编程题自动阅卷技术的研究与实现[D]. 长沙:中南大学,2007.
- [2] 张晓光,张贞,牛永洁. 对于C++语言考试编程题自动评分系统的研究[J]. 现代电子技术,2010(18):51-52.
- [3] 李富星,牛永洁. 基于XML结构的C语言考试的自动评分系统[J]. 科学计算与信息处理,2010(16):130-132.
- [4] 余石泉,周肆清. 正则表达式在编程题自动阅卷中的应用[J]. 计算机技术与发展,2007,17(7):244-246.
- [5] 马培军,王甜甜,苏小红. 基于程序理解的编程题自动评分方法[J]. 计算机研究与发展,2009(7):1136-1142.
- [6] 王倩,苏小红,马培军. 有语法错误的编程题自动评分方法研究——用局部语法分析和采分点匹配实现[J]. 计算机工程与应用,2010(17):239-242.
- [7] 郑燕娥,郑志明,陈维斌. 基于反射机制的Java编程题自动评分技术[J]. 滨州学院学报,2012(3):96-99.

(责任编辑:肖锡湘)