基于词法、语法分析的试题导入系统的研究

王 甲,康慕宁
(西北工业大学 计算机学院,陕西 西安 710072)

摘 要:针对传统在线考试系统试题录入效率低下的问题,提出了一种基于词法分析和语法分析的试题验证解析方法。阐述了试题导入系统的架构和基本原理,重点说明了词法分析和语法分析的应用原理和方法,并根据试题导入应用的特殊性,对语法分析进行改良,提出了附加栈法和错误预测捕捉法,从而提高解析能力和错误处理能力。

关键词:词法分析;语法分析;试题导入;附加栈法;错误预测捕捉法
中图分类号:TP391.43 文献标识码:B

Research on question importing system based on lexical analysis and syntax analysis

WANG Jia, KANG Mu-Ning
(Dept. of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: To deal with the low-efficiency of questions entering in online exam systems, a method of validating and parsing questions based on lexical analysis and syntax analysis was proposed. The method combined question import system with lexical analysis and syntax analysis. It illustrated the architecture and basic principle of the question import system, and the principle of lexical analysis and syntax analysis was mentioned as an emphasis. Furthermore, two methods named added-stack method and error-forecasting-catch method were proposed to improve parsing and error handling ability.

Key words: lexical analysis; syntax analysis; question importing; added-stack method; error-forecasting-catch

利用自动化的手段将已有试题文档直接导入在线考试系统,可以很大程度地减轻系统管理员的工作负担。但自动化导入对于保证录入试题的准确、保密等各方面,都有着严格要求。

本文将编译原理中的相关概念引入到试题导入系统中,提出了一种结合词法分析、语法分析等技术的试题导入系统。在该系统中,首先通过词法分析将试题文本打断为孤立的单词(token),进而将各个单词进行归类和封装;之后,利用封装后的单词对象,使用预定义的语法树进行匹配和验证;同时,单词表对整个过程涉及到的临时数据进行存储和必要支持。结果表明,此试题导入系统录入效率高、保密性强、可重复使用,用户体验极佳。

1 试题导入系统的基本设计

系统实现目标为:输入文件对象,用户控制导入,系统对被输入文件中的各个试题信息元素进行验证。如果合法,则导入数据库;否则,定位文件中不合法的信息元素,并通过GUI界面给用户提示。基于以上的需求,决定使用MVC架构(模型—视图—控制器)来部署整个系统[1]。总体架构如图1所示。

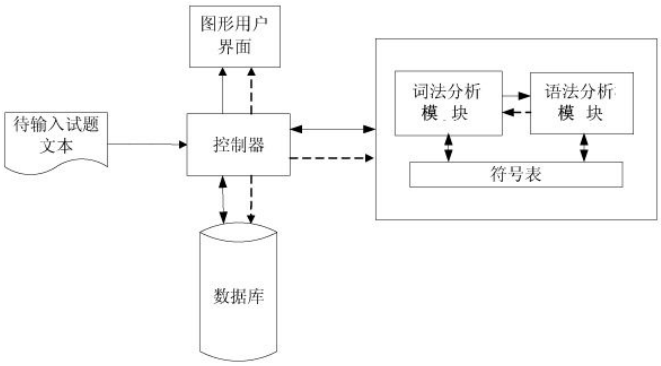

图1 试卷导入系统架构图

图1中模型模块是整个系统的核心,其中的词法、语法分析是实现模型模块的关键。

2 词法、语法分析在试卷导入系统中的应用

由于整个导入系统采用 C# 语言编写，本着减小开发规模和提高开发品质的原则，决定使用 C#Lex 和 CsCup 作为词法、语法分析器生成工具。是本文使用的语法分析程序生成工具是当今流行的自下而上的前后文无关文法 LR(1)[2]。

2.1 词法分析的应用

2.1.1 词法分析

从左到右逐个字符地读入源程序，对构成源程序的字符流进行扫描之后根据构词规则识别单词。词法分析的输入是源程序，输出是各个单词内部表示的记号流。完成一个词法分析周期需要 3 个步骤：读入一定个数的字符、对当前字符的模式识别和生成并输出单词。

2.1.2 试卷导入系统中词法分析的关键技术

(1) 对字符串的模式识别。根据正则文法相关理论，正则文法可以构造相应的正则式，而正则式可以构造对应的 DFA (有限自动机)，DFA 则可以有效地完成字符串的模式识别。即通过书写正则式来定义模式。

(2) 单词对象的生成。在完成一个元素的模式识别后，需要向语法分析程序输出已经识别的单词信息。在传统的 Lex 中，输出是以一个全局整形变量的形式提供给语法分析程序的。而针对面向对象语言的词法生成工具，都是以单词类的实例——单词对象的形式来输出。

在试卷导入程序中，单词的诸多信息，包括单词种类、行号、起始位置、内容都是有意义的，所以单词类应该包括这些信息。

2.2 语法分析的应用

2.2.1 语法分析

根据语言的语法规则，分析源程序的语法结构，即分析如何由这些单词组成各种语法范畴，并在分析过程中，对源程序进行语法检查[3]。

一个语法结构，可以用文法的形式定义：一个文法 $G[S]$ 可表示成形如 (V_n,V_t,P,S) 的四元式。其中 V_n,V_t,P 均为非空的有限集，分别称为非终结符集、终结符号集和产生式集。$S\in V_n$ 为文法的开始符号。

为说明问题，这里形式化地描述一道基本形式的选择题。规定基本形式的选择题有以下特征：(1) 阿拉伯数字形式的试题序号；(2)分值；(3)题干；(4)若干个选项；(5)答案 。如图 2 所示的试题就符合该特征。

图 2 符合基本特征的选择题

根据以上的规定和正则表达式的使用方法，可以确定选择题的集合即终结符号集如表 1 所示。

表 1 终结符号与正则表达式的对应关系

特征	示例	正则表达式	标识符号
试题序号	1	^[\s\t]*[0-9]+[..]	tnumber
分值	10 分	[(][0-9]+(分)[)]	tpoint
题干	Dos 目录是()	[^\n\r]+	tmain
选项	A.菜单结构 B.树形结构	^[\s\t]*[A-Za-Z]+[..] [^\n\r]+	toption
答案	答案:B	^[\s\t]* 答案[:：] [^\n\r]+	tanswer

之后，可以从基本形式的选择题的特征和表 2 的标识符号，构造文法的 V_n 和 P 集合：

V_n={Number,Point,Main,Option,Answer,Questions,SelectQ,Infor}

P= {Questions:Questions SelectQ | SelectQ

,SelectQ:Infor Option Answer | Infor Answer Option

,Infor:Number Main Point | Number Point Main | Point Number Main

,Option:Option toption | toption

,Number:tnumber,Point:tpoint,Main:tmain,Answer:tanswer}

显然，该文法的开始符号 S 为 Questions 。

该文法 $G(S)$ 可以完整表示所定义的选择题，并兼容多个此类选择题的集合、不定选项、分数、试题号、题干若干种不同出现顺序、答案、选项乱序的情况。而不符合该文法的则不接受。

在试卷导入系统中，语法分析主要完成 3 个功能：根据预定义的语法对词法分析输出的单词流进行验证；根据语法对输入错误进行捕捉；对于验证成功的输入生成试题对象。

2.2.2 单词流验证

对于一个确定的文法 $G(S)$，单词流验证的核心步骤是根据词法分析得来的终结符 $t\in V_t$，自下而上地构造一个语法树。其中该语法树有以下特征：叶子节点仅为单词 $t\in V_t$；根节点仅为 S；对任意一个父节点和其子节点，存在推导式 $p\in P$，且 p 的左部分为父节点，右部分为子节点。

例如，对于上例中的试题和文法 $G(S)$，可以确定词法分析返回的单词流为：

tnumber-tpoint-tmain-toption-toption-tanswer (1)

此时可以构建一个如图 3 所示的符合上述特征的语法树。

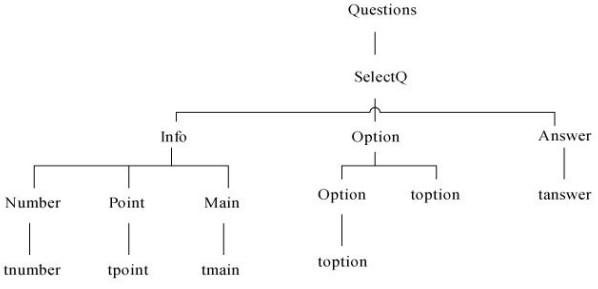

图 3 对应单词流所构建的语法树

对任意一个单词流和文法 $G(S)$,该单词流为该文法的一个句子,当且仅当单词流中的各个元素能通过上述规则构建一个语法树。如果单词流经验证是文法的句子,对于试卷导入系统来说,此时验证成功;反之,需要进行错误捕捉和错误处理。

2.2.3 试题对象的产生

规约过程将会进行弹出栈顶若干元素,规约成某非终结符,然后将此非终结符压入当前分析栈的一系列动作。伴随这个动作的完成,被弹出栈的若干个符号的细节信息将会丢失。因此,必须使用辅助手段来记录被规约的符号,才可以有效地解决这个问题。

为了实现边验证边生成试题的目的,这里提出一种“附加分析栈”的方法来实现。

附加分析栈方法:设立一个栈以存储各种分析的中间结果,对当前分析栈进行有限度的跟踪。附加栈的运作方式为:

(1) 移进动作,附加分析栈不进行任何操作;

(2) 规约动作,则将规约动作对应的表达式右边的部分封装成一个预定义的对象,并压入附加栈。

例如,一个上文中的 $G(S)$如果有一个单词流为

tnumber-tpoint-tmain-toption-toption-tanswer (2)

则附加栈的作用如表 2 所示。

表 2 附加栈分析(2)单词流时的运行过程

分析栈	对应表达式	动作	模拟分析栈
tnumber		进栈	
Number	Number: tnumber	规约	Obj_Number
tpointNumber		进栈	Obj_Number
PointNumber	Point: tpoint	规约	Obj_Point Obj_Number
tmainPointNumber		进栈	Obj_Point Obj_Number
MainPoint Number		规约	Obj_MainObj_PointObj_Number
Infor	Infor: Number Point Main	规约	Obj_Infor
…	…	…	…

在表 2 中,需要注意两个分析栈从上到下依次为栈顶、栈中、栈底;Obj_Number,Obj_Point,Obj_Mian 对象为词法分析输出类的实例;Obj_Infor 对象为 Obj_Number,Obj_Point,Obj_Mian 三元组构成的对象。

一个成功的解析会生成 Questions 对象,通过控制器调用存储过程,可以存入数据库。

2.2.4 错误捕捉机制

试卷中可能的错误一般有 3 种:元素重复、元素缺失、其他未知错误。对于前两种错误,由于其可预知的特性,这里提出一种“预测捕捉法”进行错误处理。

预测捕捉法:根据试题文件中可能的典型错误单词流 W,扩充文法 $G(S)$中的产生式集合 P,使得该错误单词流 W 也可以根据上述规则 $P \vee Pe$ 构建一个语法树,其中 Pe 为预测捕捉产生式集合。完成构建后,则可以记录该错误,并继续剩下的单词流分析。预测捕捉产生式一般由以下的原则产生:对于包含非终极符 A,终结符 a,产生式 $A:a$ 的文法,元素 a 重复的错误,可以用 $A:A\ a \mid a$ 的形式进行扩展;元素 a 缺失的错误,可以用 $A:\varepsilon$ 的形式进行扩展。

例如,如果上述例题中“1.(10 分) Dos 目录是()”被错写成“1. 1.(10 分) Dos 目录是()”或“ (10 分) Dos 目录是()”,此时,原先文法 $G(S)$将无法成功构建一个语法树。但是,如果给 P 加入一个“Number: Number tnumber”或“Number: ε”的产生式,就能够捕捉到这样的错误情况。

其他形式的错误,因为其不可预知,采用特殊的单词 error 进行捕捉。error 单词是语法生成工具中普遍预定义的单词,其基本原理是:遇到不可辨识的栈顶状态和当前单词,它从栈中丢弃状态和对象直到回到一个可以接受 error 的状态,error 单词被移进,之后语法分析程序不断地读进单词,如果可以接受则在分析栈中压进,否则丢弃。

在语法生成工具 CsCup 中,通过列举表达式集 P 可以很方便地生成一个接受表达式集 P 所表示文法句子的 C# 源文件。CsCup 源文件书写格式和 error 单词使用方法请参阅参考文献[5-6]。

本文着重介绍了词法、语法分析在试卷导入中应用的步骤和关键技术。实践证明,该方法具有强大的错误捕捉和错误处理能力。同时,可以通过添加新的表达式,扩展所支持的题目类型,因此具有良好的可扩展性。另外,对其他涉及到数据验证的应用,本文所介绍的方法也具有适用性。

参考文献

[1] ERIC F,ELISABETH F,KETHY S.Head First 设计模式 (中文版)[M].北京:中国电力出版社,2007.

[2] LEVINE J R,MASON T,BROWN D.lex & yacc, Second Edition [M].美国, O'Reilly Media,1992.

[3] 蒋立源,康慕宁.编译原理[M].西安.西北工业大学出版社,2005.

(下转第 70 页)