

编译原理课程项目



编译原理课程项目

一、课程介绍

二、项目任务

三、项目进度安排计划

一、编译原理课程项目课程介绍

课程性质：必修课

必修：每个同学必须修读并完成相应的设计要求。

二、项目任务

1. 按课程项目设计任务书的具体要求进行完成。
2. 项目设计报告书要按 《项目设计报告书 版头及正文格式》 文件中的格式书写。
3. 完成方法：每个学生独立完成。
4. 上交内容：项目设计源程序、项目设计报告书、使用说明书、项目的可执行程序、测试文本及测试源程序、测试报告、项目报告讲解视频
5. 项目进度安排。
6. 所设计的项目软件系统应该是一个windows应用程序（即windows界面）。

实验回顾

第一个

实验二：生产流水线的自动设计器

- 一、实验内容：

输入一个表示要准备进行设计的生产流水线对应的正则表达式，最终完成整个生成流水线的自动设计。

- 二、必做实验要求：

(1) 正则表达式应该支持单个字符，运算符号有：连接 选择(|) 闭包(*) 正闭包(+) 可选(?) 括号

(2) 要提供一个源程序编辑界面，让用户输入表示生成流水线处理过程的正则表达式(可保存、打开正则表达式文件)

(3) 需要提供窗口以便用户可以查看转换得到的NFA(用状态转换表呈现即可)

(4) 需要提供窗口以便用户可以查看转换得到的DFA(用状态转换表呈现即可)

(5) 需要提供窗口以便用户可以查看转换得到的最小化DFA(用状态转换表呈现即可)

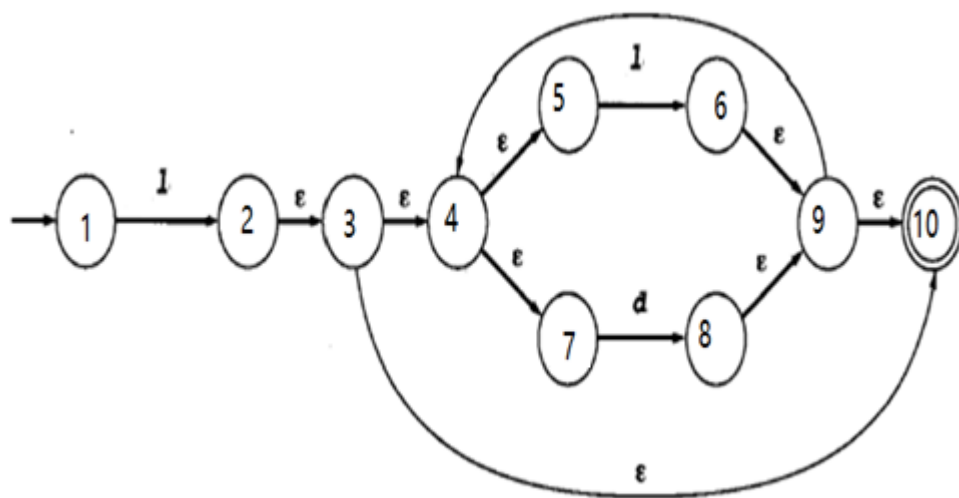
实验二

1.输入正则表达式:

如: $l(l|d)^*$

2.正则表达式 \rightarrow NFA

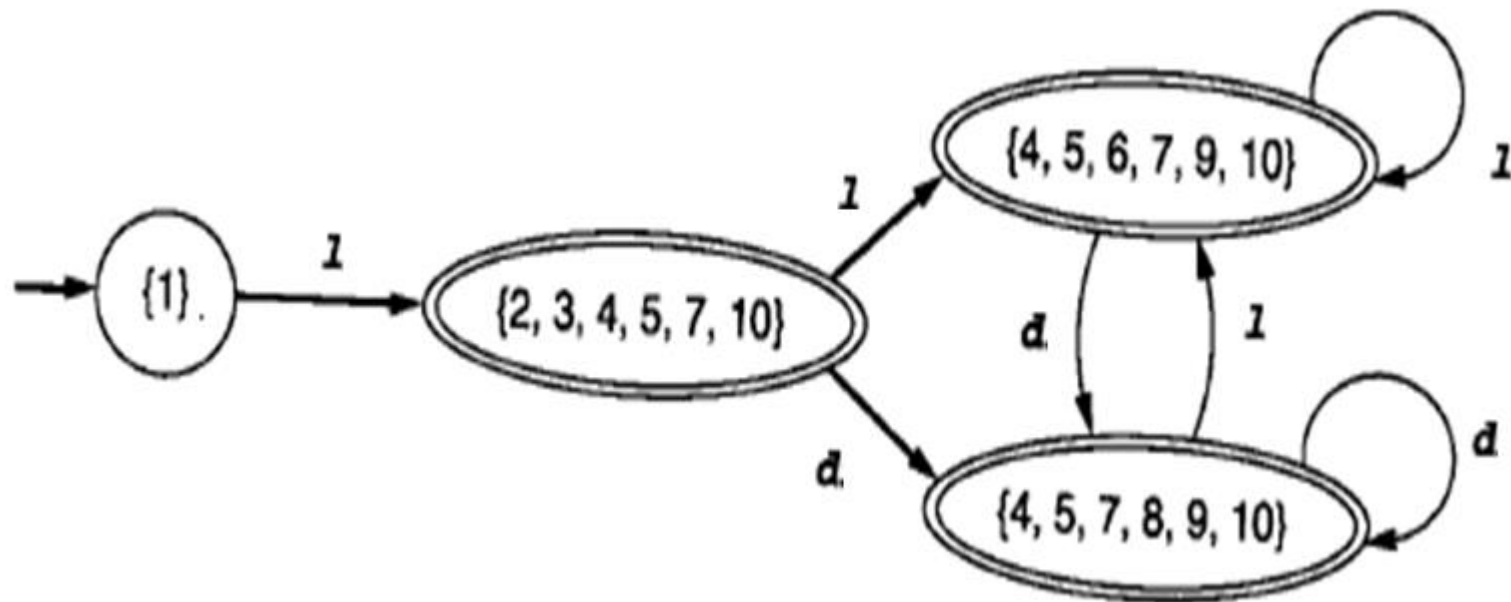
输出：如下图的NFA或该图对应的状态转换表



		1	d	#
—	1	2		
	2			3
	3			4, 10
	4			5, 7
	5	6		
	6			9
	7		8	
	8			9
	9			4, 10
+	10			

3.NFA→DFA

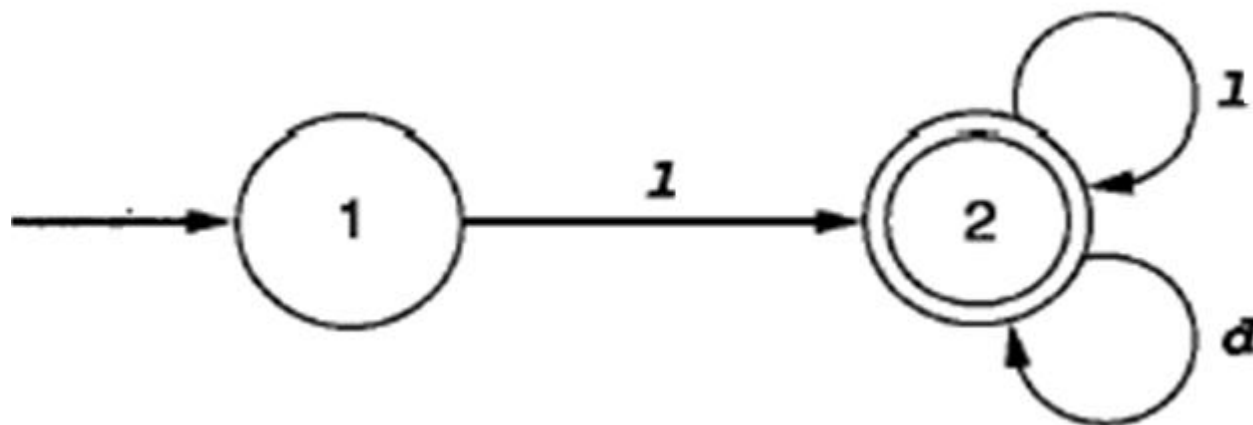
输出：如下图的DFA或对应的状态转换表(如下表)



状态集合 \ 字符	1	d
{ 1 }	{ 2, 3, 4, 5, 7, 10 }	
{ 2, 3, 4, 5, 7, 10 }	{ 6, 9, 4, 7, 10, 5 }	{ 8, 9, 4, 7, 5, 10 }
{ 6, 9, 4, 7, 10, 5 }	{ 6, 9, 4, 7, 10, 5 }	{ 8, 9, 4, 7, 5, 10 }
{ 8, 9, 4, 7, 5, 10 }	{ 6, 9, 4, 7, 10, 5 }	{ 8, 9, 4, 7, 5, 10 }

4.DFA最小化

输出：如下图的最小化DFA或对应的状态转换表



—

+

	l	d
1	2	
2	2	2

课程项目的第一项任务

- 设计一个应用软件，以实现输入某一高级程序设计语言的所有单词的正则表达式-->NFA--->DFA-->DFA最小化-->词法分析程序。

课程项目的第一项任务要求

具体要求：

(1) 以文本文件的方式输入某一高级程序设计语言的所有单词对应的正则表达式，系统需要提供一个操作界面，让用户打开某一语言的所有单词对应正则表达式文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。

(2) 需要提供窗口以便用户可以查看转换得到的NFA（可用状态转换表呈现）

(3) 需要提供窗口以便用户可以查看转换得到的DFA（可用状态转换表呈现）

(4) 需要提供窗口以便用户可以查看转换得到的最小化DFA（可用状态转换表呈现）

(5) 需要提供窗口以便用户可以查看转换得到的词法分析程序（该分析程序需要用C语言描述）

(6) 对要求(5)得到的源程序进行编译生成一个可执行程序，并以该高级程序设计语言的一个源程序进行测试，输出该该源程序的单词编码。

(7) 对系统进行测试：(A) 先以TINY语言的所有单词的正则表达式作为文本来测试，生成一个TINY语言的词法分析源程序；(B) 接着对这个词法分析源程序利用C/C++编译器进行编译，并生成可执行程序；(C) 运行该可执行程序，以sample.tny来测试，输出该TINY语言源程序的单词编码文件sample.lex

(8) 要求应用程序为Windows界面

(9) 书写完善的软件文档

第一项任务要求的测试数据

- Tiny语言

TINY 语言的单词分类

(1). 关键词:

if then else end repeat until read write 不区分大小写

(2). 专用符号: + - * / < = { } ; :=

(3). identifier和number, 则定义:

identifier: 字母开头, 后面可跟若干个字母, 且不区分大小写

number: 数字符号开头, 后面可跟若干个数字

也可以抽象地表示为:

identifier = letter letter*

number = digit digit*

letter = a|..|z|A|..|Z

digit = 0|..|9

(4). 空格由空白、换行符和制表符组成。

(5). 用表示{ }注释, 注释可以放在任何空白出现的位置(即注释不能放在标记内)上, 且可以超过一行。注释不能嵌套。

1.TINY语言词法测试结果

- 1.打开包含有所有TINY语言单词的正则表达式的文本文件
- 2.生成对应的NFA图，需要界面可以查看该图
- 3.生成对应的DFA图，需要界面可以查看该图
- 4.生成对应的最小DFA图，需要界面可以查看该图
- 5.把最小DFA图转换为对应的词法分析源程序，该源程序需要在C/C++编译器环境下成功编译，需要界面可以查看词法分析源程序
- 6.利用这个词法分析源程序进行词法分析，并生成单词编码文件，需要界面可以查看单词编码文件内容。

第二个

实验三：文法问题处理器

一、实验内容：

设计一个应用软件，以实现文法的化简及各种问题的处理。

二、实验要求：

1.必做内容

(1) 系统需要提供一个文法编辑界面，让用户输入文法规则（可保存、打开存有文法规则的文件）

(2) 化简文法：检查文法是否存在有害规则和多余规则并将其去除。系统应该提供窗口以便用户可以查看文法化简后的结果。

(3) 检查该文法是否存在左公共因子（可能包含直接和间接的情况）。如果存在，则消除该文法的左公共因子。系统应该提供窗口以便用户可以查看消除左公共因子的结果。【从某个非终结符号开始只需要消除做4次推导内就可以发现的左公共因子，如果超过4次推导还存在左公共因子就直接产生出错处理即可】

(4) 检查该文法是否存在左递归（可能包含直接和间接的情况），如果存在，则消除该文法的左递归。系统应该提供窗口以便用户可以查看消除左递归后的结果。

(5) 求出经过前面步骤处理好的文法各非终结符号的first集合与follow集合，并提供窗口以便用户可以查看这些集合结果。【可以采用表格的形式呈现】

课程项目的第二项任务要求

- 设计一个应用软件，以实现输入某一高级程序设计语言的所有语法对应的BNF文法-->化简--->消除文法中存在的左公共因子及左递归-->求出改造后文法的非终结符号的First集合和Follow集合-->生成LL(1)分析表-->以LL(1)语法分析方法进行语法分析并生成相应的语法树。

课程项目的第二项任务

具体要求：

- (1) 以文本文件的方式输入某一高级程序设计语言的所有语法对应的BNF文法，因此系统需要提供一个操作界面，让用户打开某一语言的所有语法对应的BNF文法的文本文件，该文本文件的具体格式可根据自己实际的需要进行定义。
- (2) 需要提供窗口以便用户可以查看文法化简后的结果（可用表格形式进行呈现）
- (3) 需要提供窗口以便用户可以查看消除左公共因子和左递归之后的新文法（可用表格形式进行呈现）
- (4) 求出改造后文法的每个非终结符号的First集合和Follow集合，并需要提供窗口以便用户可以查看该结果（可用两张表格的形式分别进行呈现）
- (5) 构造出LL(1)分析表，并需要提供窗口以便用户可以查看该结果（可用表格形式进行呈现）
- (6) 采用LL(1)语法分析方法进行语法分析并生成相应的语法树，每个语句的语法树结构可根据实际的需要进行定义。（语法树需要采用树状形式进行呈现）
- (7) 以TINY语言的所有语法以及第一项任务的测试结果sample.lex作为测试，并生成对应的语法树并呈现出来。
- (8) 要求应用程序为Windows界面
- (9) 书写完善的软件文档
- (10) 选做内容：可以生成中间代码。

第二项任务要求的测试数据

- Tiny语言

2.TINY语言的所有语法

BNF中的TINY的文法

program \rightarrow *stmt-sequence*
stmt-sequence \rightarrow *stmt-sequence* ; *statement* | *statement*
statement \rightarrow *if-stmt* | *repeat-stmt* | *assign-stmt* | *read-stmt* | *write-stmt*
if-stmt \rightarrow **if** *exp* **then** *stmt-sequence* **end**
 | **if** *exp* **then** *stmt-sequence* **else** *stmt-sequence* **end**
repeat-stmt \rightarrow **repeat** *stmt-sequence* **until** *exp*
assign-stmt \rightarrow **identifier** := *exp*
read-stmt \rightarrow **read** **identifier**
write-stmt \rightarrow **write** *exp*
exp \rightarrow *simple-exp* *comparison-op* *simple-exp* | *simple-exp*
comparison-op \rightarrow < | =
simple-exp \rightarrow *simple-exp* *addop* *term* | *term*
addop \rightarrow + | -
term \rightarrow *term* *mulop* *factor* | *factor*
mulop \rightarrow * | /
factor \rightarrow (*exp*) | **number** | **identifier**

课程项目的第三项任务

前两项任务都完成后...

- mini-c语言作为测试
- (1) 以mini-c的词法进行测试，并以至少一个mini-c源程序进行词法分析的测试。
- (2) 以mini-c的语法进行测试，并以测试步骤(1)的源程序所生成的单词编码文件进行语法分析，生成对应的语法树。

第三部分 项目进度安排

时间	进度安排
第 1 周	任课老师课程项目介绍
第 1 周~第 5 周	完成第一个项目任务的软件设计及文档书写要求，并以 Tiny 语言的词法进行测试及书写测试报告
第 6 周~第 10 周	完成第二个项目任务的软件设计及文档书写要求，并以 Tiny 语言的语法进行测试及书写测试报告
第 11 周~第 13 周	<ol style="list-style-type: none">1. 以 Mini-C 进行整个项目的测试（包含词法和语法的测试），并书写测试报告2. 完成最后实验报告的整合和自评3. 完成系统使用说明书的书写4. 项目报告讲解视频的录制（15 分钟）
第 14 周	<p>完成如下内容：</p> <ol style="list-style-type: none">1. 整理好项目设计源程序、测试文本及测试源程序、测试报告、项目的可执行程序、使用说明书、项目设计报告书和讲解视频。2. 第 14 周周五晚上 12 点之前提交上述所有资料，在砺儒云课堂进行提交。
第 15~17 周	<ol style="list-style-type: none">1. 抽查、评分。2. 抽查到的同学需到学院实验室解释相关的工作。