

# AutomatonConversion项目说明文档

## 初稿

计研091 黄崇迪 2009210940

huangcd.thu@gmail.com

January 4, 2010

## 目录

<b>1</b>	<b>项目说明</b>	<b>2</b>
1.1	项目目标及实现 . . . . .	2
1.2	项目开发环境 . . . . .	2
1.3	项目依赖类库 . . . . .	2
<b>2</b>	<b>程序使用说明</b>	<b>3</b>
2.1	基本使用 . . . . .	3
2.2	动态接受字符串 . . . . .	4
<b>3</b>	<b>程序实现介绍</b>	<b>4</b>

# 1 项目说明

## 1.1 项目目标及实现

所选项目为**NFA/DFA的转换工具**。具体要求如下：

1. 用Java编写，保证可扩展性。
2. 检查某个字符串是否被指定的自动机接受。
3. 判断语言是否为空，是否为无穷。
4. 实现NFA到DFA的相互转换。

目前除了实现上述要求内容以外，额外实现了一些功能，具体包括：

1. XML到自动机的相互转换
2. 自动机的可视化及动态接受字符串的过程

## 1.2 项目开发环境

操作系统	Windows Vista <sup>TM</sup> Home Premium Server Pack 2
CPU	Intel(R) Core(TM)2 Duo CUP P7350 @ 2.00GHz × 2
内存	2.00 GB
JDK版本	1.6.0_13
开发工具	IntelliJ IDEA 8.1.4

## 1.3 项目依赖类库

项目依赖于JUNG 2.0和Dom4j 1.6两个类库。最多包含下列jar文件：

dom4j.jar collections-generic-4.01.jar jung-visualization-2.0.jar  
jung-algorithms-2.0.jar jung-graph-impl-2.0.jar jung-api-2.0.jar  
concurrent-1.3.4.jar stax-api-1.0.1.jar colt-1.2.0.jar wstx-asl-3.2.6.jar

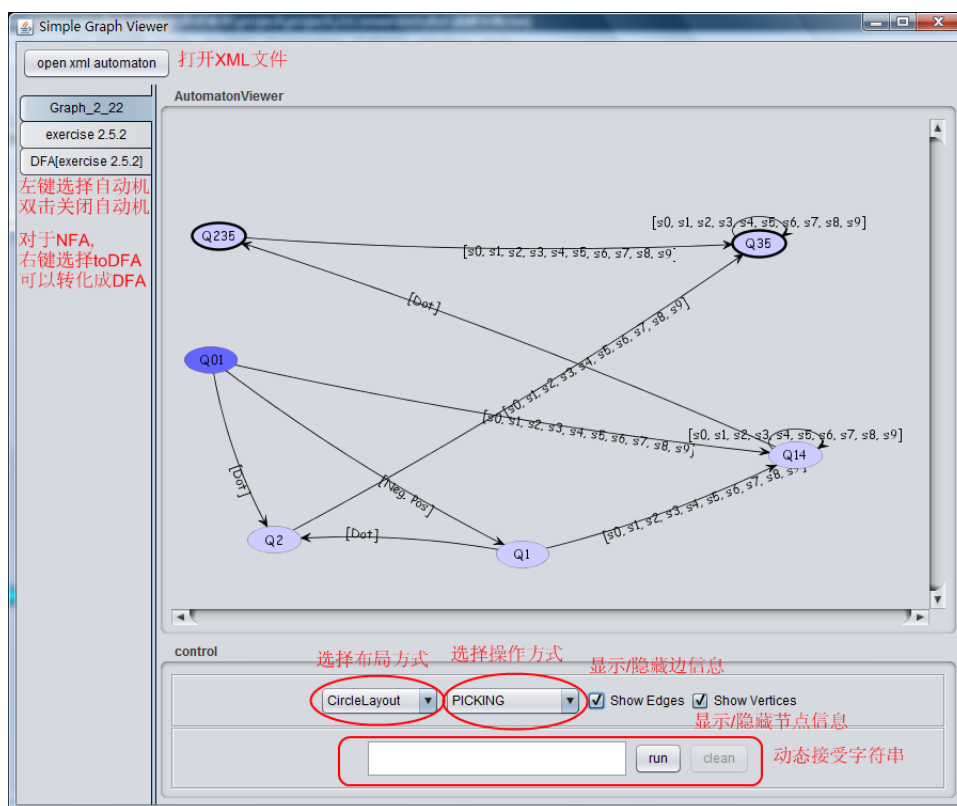


Figure 1: 程序主题界面及简要使用说明

## 2 程序使用说明

### 2.1 基本使用

通过bin目录下面的run.bat文件启动程序。<sup>1</sup>启动后通过open xml automaton按钮打开XML文件。图1是程序运行某个时刻的截图及各个按钮的简要使用说明。

AutomatonViewer部分显示自动机，初始状态用深蓝色表示，终止状态用粗框表示，其他状态用浅蓝色表示。转换以边的形式表示（如果一个状态可以通过多个符号转换成另一个状态，那么图中把多个转移合成为一

<sup>1</sup>程序目前只经过非常简单的测试，错误可能比较多。如有任何问题，请联系 huangcd.thu@gmail.com

条边)。

AutomatonViewer部分支持鼠标中键缩放，图形拖动（TRANSFORMING状态）和节点拖动（PICKING状态）等功能。鼠标在节点或者边上悬停出现详细信息。也可以选择不同的布局方式（但目前没有效果比较好的布局方式）。

目前程序的功能主要针对DFA实现。因此对于NFA，动态接受字符串部分不可用，但可以通过右键点击标签选择toDFA选项来把NFA转换成DFA。

## 2.2 动态接受字符串

目前动态接受字符串只针对DFA实现。使用的方法是在输入框中输入字符串（[字符之间需要用“|”分割](#)），然后点run执行。

接受过程中，当前状态在AutomatonViewer中显示为黄色，当前符号在输入框中显示为蓝色。如果字符串被拒绝，状态和符号都被显示为红色。如果字符串被接受，接受的状态被显示为绿色。

## 3 程序实现介绍

### 自动机的表示

自动机在程序里面被表示成状态的集合。转移被表示为状态的属性。程序里面使用FiniteAutomaton表示抽象的自动机，DFA和NFA继承FiniteAutomaton分别表示相应的自动机。状态同样对应地有DFAS-tate和NFAS-tate（实现State接口）。

### NFA到DFA的转换

根据课本上的算法实现，从初始状态开始根据传递闭包的转移逐步构造整个DFA。在NFA.toDFA()函数中实现。

### 判断自动机是否为空

通过广度优先搜索检查从开始状态是否可达终止状态。在 `DFA.isEmpty()` 函数中实现。对于NFA，通过 `NFA.toDFA().isEmpty()` 实现。

### **判断自动机是否为无穷**

通过深度优先搜索检查从开始状态到终止状态的路径中是否存在环。在 `DFA.isInfinite()` 函数中实现。对于NFA，通过 `NFA.toDFA().isInfinite()` 实现。

### **判断字符串时候被接受**

根据输入符号遍历状态，字符串结束的时候恰好到达终结状态，说明字符串被接受。在 `DFA.accept()` 函数中通过调用 `DFAState.shift()` 实现。对于NFA，同样通过 `NFA.toDFA().accept()` 实现。

### **XML和自动机的相互转换**

根据FA.dtd的描述，调用dom4j进行解析或写出。具体实现在 `automaton.io.xml`包中的几个类。

### **显示**

利用jung实现，具体实现见graph包和ui包。

### **简单使用例子**

见 `sample.Test`。

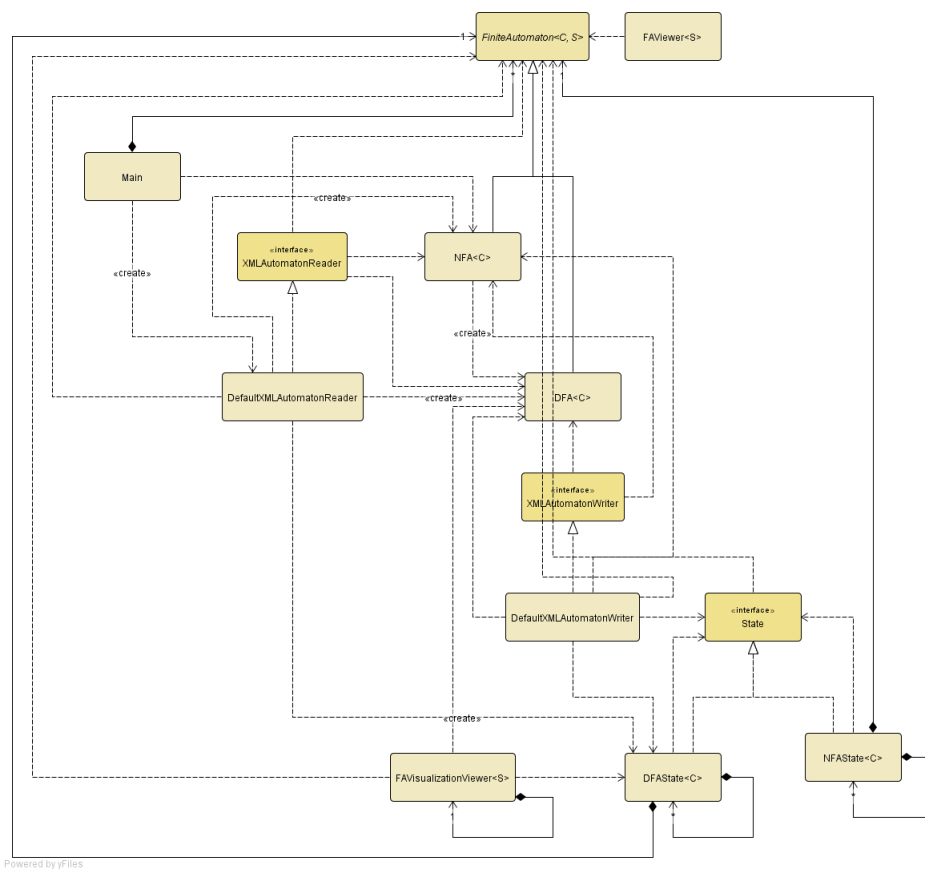


Figure 2: 各个类之间的关系