JFLAP2Ti*k*Z

Andrew Mertz and William Slough

Introduction

JFLAP[1] is a popular Java program for experimenting with finite state machines, Turing machines and other concepts from Formal Languages and Automata Theory. Version 7 of JFLAP can export JPG, PNG, GIF, BMP, or SVG images. JFLAP2Ti*k*Z is a Groovy[2] script that converts a JFLAP jff file representing a finite automaton, pushdown automaton, or Turing machine into a LaTeX file depicting the automaton graphically using Ti*k*Z.

Requirements

To use JFLAP2Ti*k*Z you will need Java installed. Additionally, you may find it useful to have Groovy installed as well. You will need to download either `jflap2tikz.jar` (if you only have Java installed) or `jflap2tikz.groovy` (if you also have groovy installed).

Usage

JFLAP2Ti*k*Z is invoked from the command line with:

```
java -jar jflap2tikz.jar -i example.jff
```

or:

```
groovy jflap2tikz.groovy -i example.jff
```

In the above cases the output will be written to the console. Use the -h option to see the full usage information, which is also given in Figure 1.

Examples

The following figures show machines from the book JFLAP: An Interactive Formal Languages and Automata Package being converted to Ti*k*Z using JFLAP2Ti*k*Z. Note that JFLAP2Ti*k*Z currently ignores JFLAP state annotations. Furthermore, not all JFLAP files will be converted perfectly. You may have to adjust the scale and/or gridsize, or edit the resulting Ti*k*Z code to achieve the effect you want. However, JFLAP2Ti*k*Z should provide a good starting point.

[1] www.jflap.org
[2] groovy.codehaus.org

```
usage: jflap2tikz [options]
Version 1.2
 -d,--accepting-distance <distance>   Distance, in pt, between the circles of an
                                      accepting state (default is 2)
 -g,--grid <size>                     Round positions so that they are on a
                                      grid. If a size is given it sets the
                                      spacing of the grid (default is 20.0)
 -h,--help                            Show usage information and quit
 -i,--input-file <filename>           Name of a JFLAP jff file representing a
                                      finite automaton, pushdown automaton, or
                                      Turing machine. If a file is not given
                                      standard input will be used.
 -k,--keep-names                      Use the state names from the JFLAP file.
                                      The default is to replace the state names
                                      with names of the form '$q_{id}$', where
                                      id is the unique state number. Note state
                                      names will not be sanitized and thus may
                                      contain invalid TeX.
 -l,--arrow-length <length>           Length of arrows in points (default is 9)
 -o,--output-file <filename>          Name of a file for writing output. If this
                                      file already exists it will be
                                      overwritten.
 -r,--rotate                          Rotate labels along edges
 -s,--scale <x>                       1 pixel in JFLAP = x points in LaTeX
                                      (default is 1.0)
 -w,--arrow-width <width>             Width of arrows in points (default is 6)
```

Figure 1: Usage information

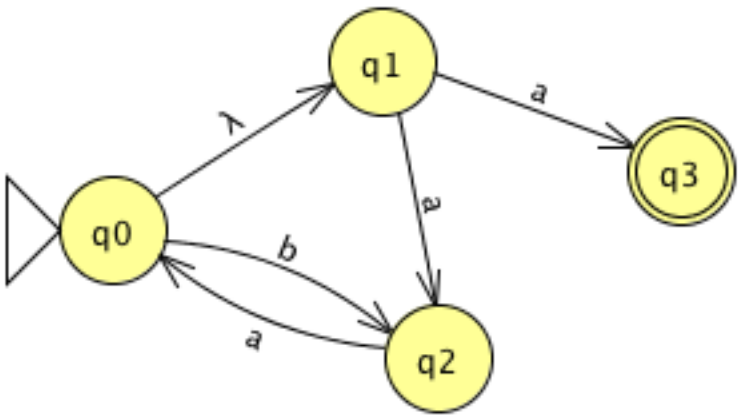

Figure 2: `ex0.1a.jff`

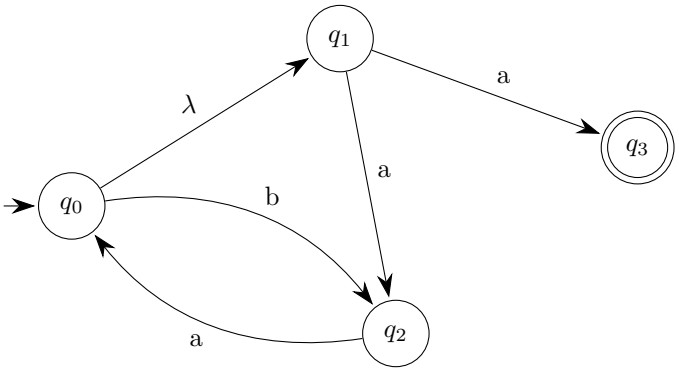

Figure 3: `ex0.1a.jff` converted to Ti*k*Z using default values

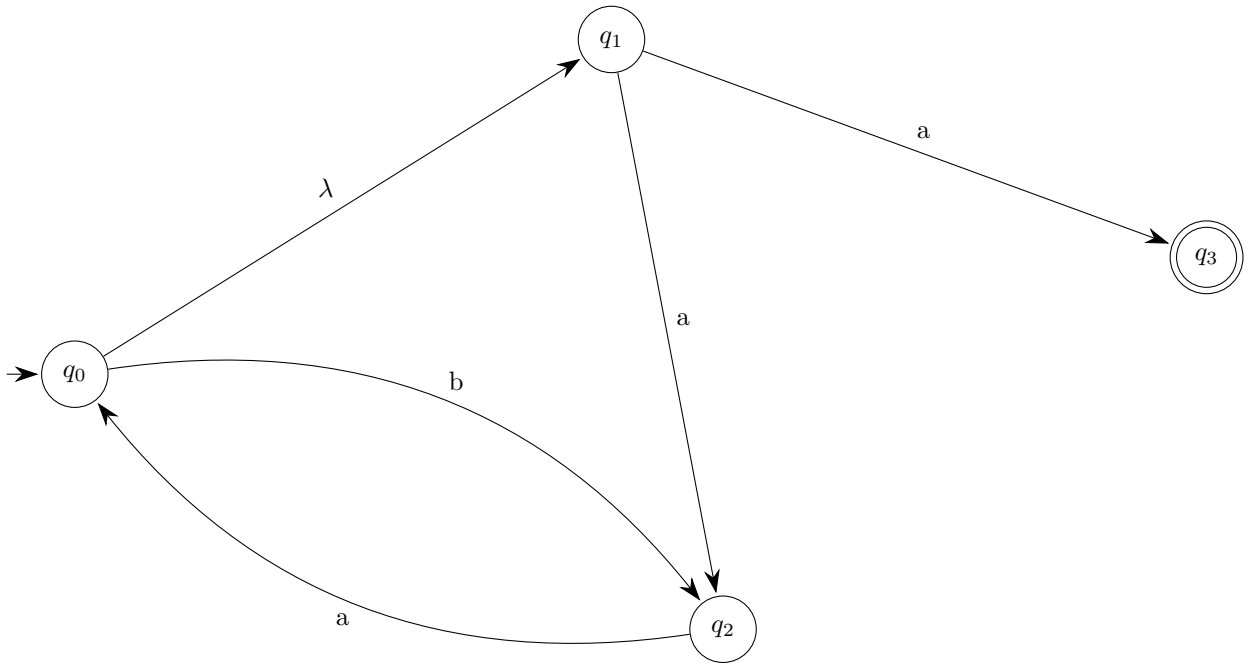

Figure 4: `ex0.1a.jff` converted to Ti*k*Z using a scale of 2

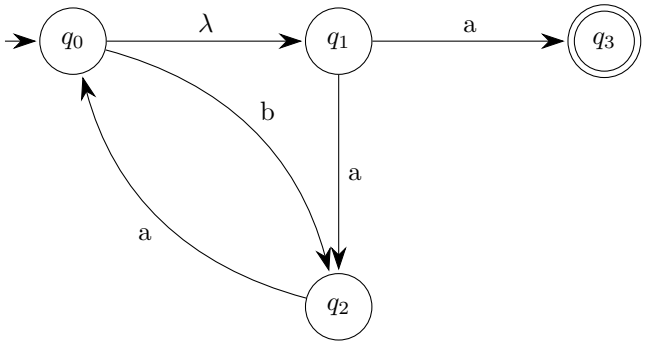

Figure 5: `ex0.1a.jff` converted to Ti*k*Z using a gridsize of 100

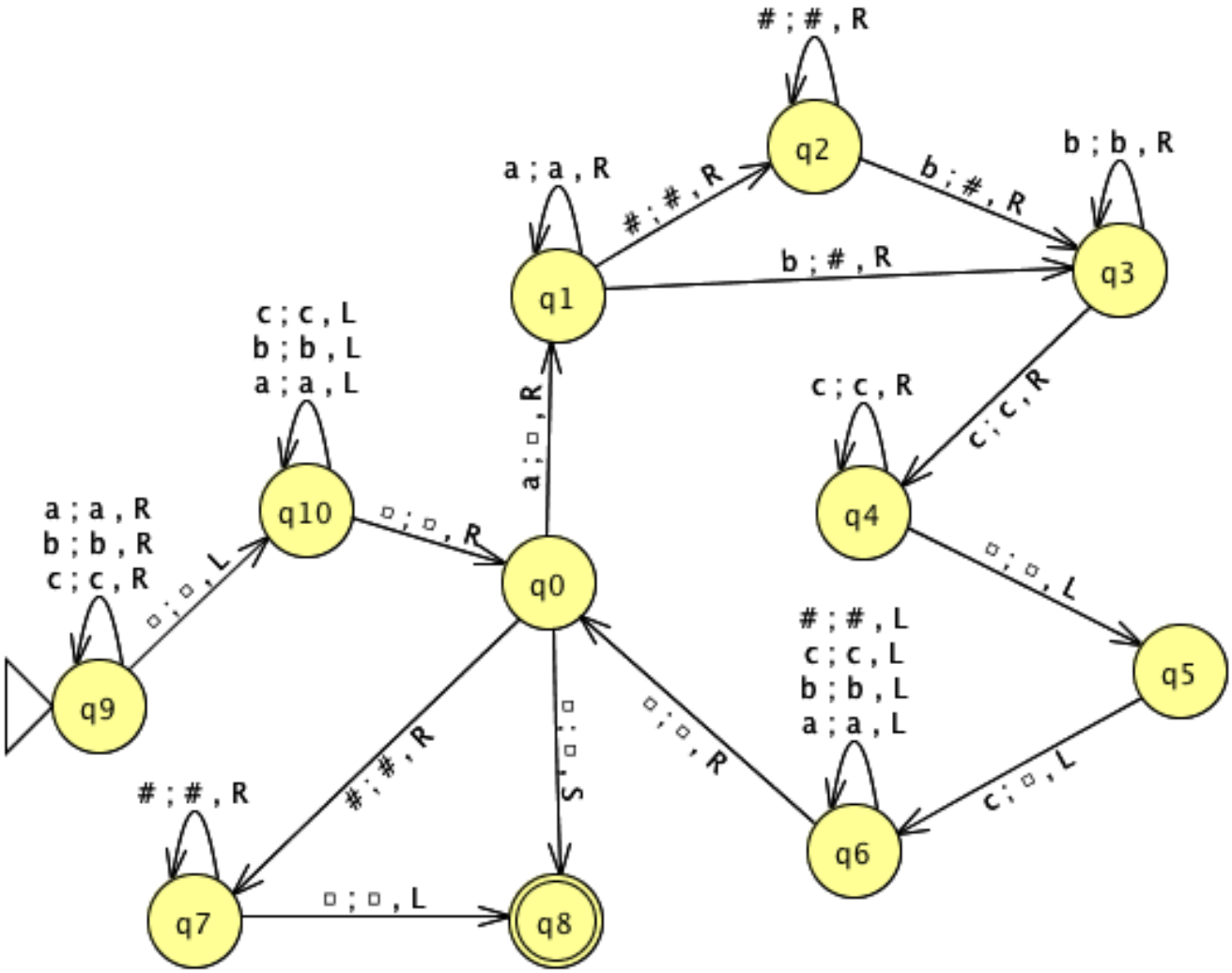

Figure 6: `ex9-anbncn.jff`

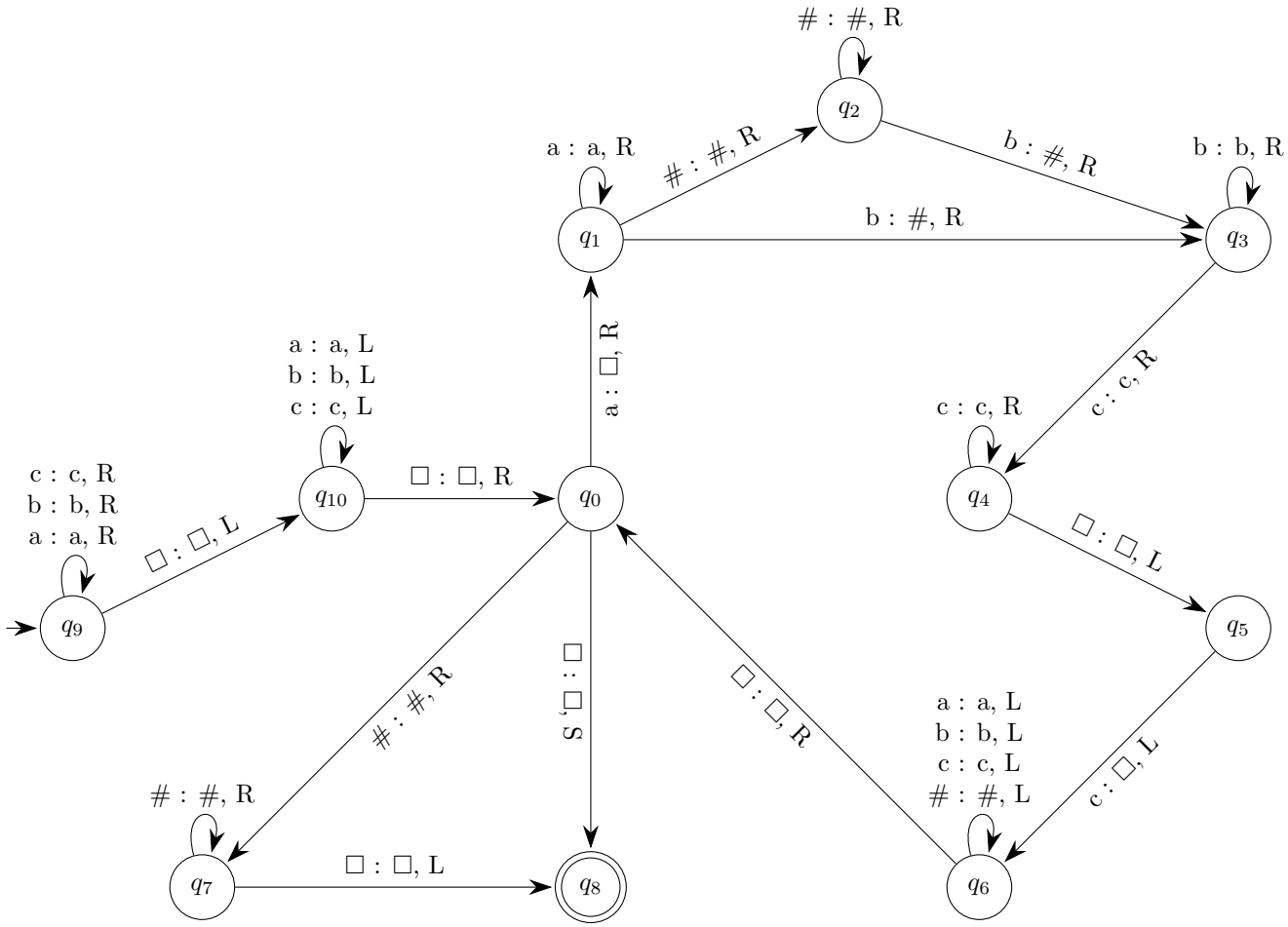

Figure 7: `ex9-anbncn.jff` converted to Ti*k*Z using a gridsize of 50 and label rotations on

License

JFLAP2Ti*k*Z and its documentation are licensed under an MIT style license