1 Introduction

"\LaTeX{} (pronounced lay-tek) is an open-source, multiplatform document preparation system for producing professional-looking documents... It is particularly suited to producing long, structured documents, and is very good at typesetting equations" [University of Edinburgh Information Services, 2014].

The capabilities of the system are greatly enhanced with the help of native and third-party packages.\footnote{The Comprehensive \TeX{} Archive Network (CTAN) is the central place for all kinds of material around \TeX{}. \url{https://www.ctan.org/?lang=en}} TikZ\footnote{\url{https://www.ctan.org/pkg/pgf?lang=en}} is a package for drawing all kinds of graphics.

This tutorial introduces the reader to \LaTeX{} and the TikZ package, particularly for drawing state diagrams of finite automata.

2 Setting up \LaTeX{}

To proceed with the tutorial, a working \LaTeX{} setup is necessary. You may install it locally on your machine, but the simplest thing to do is use Share\LaTeX{} (\url{sharelatex.com}). If you sign up using your \texttt{nd.edu} address, you’ll get unlimited private project. For further information regarding setup, visit \url{http://www.latex-project.org/get/}.

3 The preamble

Every \LaTeX{} document starts with a \textit{preamble}. To make our automata look like the ones in the textbook [Sipser, 2012], use the following preamble:

\begin{verbatim}
\documentclass{article} % What kind of document this is
\usepackage{tikz} % Import the tikz package
\usetikzlibrary{automata} % Import library for drawing automata
\usetikzlibrary{positioning} % ...positioning nodes
\usetikzlibrary{arrows} % ...customizing arrows
\tikzset{node distance=2.5cm, % Minimum distance between two nodes. Change if necessary.
ev...
4 Basics

While there are many tutorials online, I suggest two: University of Edinburgh Information Services [2014] and https://www.latex-tutorial.com/.

Here are some symbols often used in this course:

<table>
<thead>
<tr>
<th>symbol</th>
<th>control sequence</th>
<th>usual meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Σ</td>
<td>\Sigma</td>
<td>alphabet</td>
</tr>
<tr>
<td>Γ</td>
<td>\Gamma</td>
<td>another alphabet</td>
</tr>
<tr>
<td>ε</td>
<td>\varepsilon</td>
<td>empty string</td>
</tr>
<tr>
<td>o</td>
<td>\circ</td>
<td>concatenation</td>
</tr>
<tr>
<td>#</td>
<td>\texttt{#}</td>
<td>marker symbol</td>
</tr>
<tr>
<td>$</td>
<td>\texttt{$}</td>
<td>marker symbol</td>
</tr>
<tr>
<td>\textvisiblespace</td>
<td>blank symbol</td>
<td></td>
</tr>
<tr>
<td>{}</td>
<td>{ }</td>
<td>delimiters for sets</td>
</tr>
<tr>
<td>∅</td>
<td>\emptyset</td>
<td>empty set</td>
</tr>
<tr>
<td>≠</td>
<td>\neq</td>
<td>is not equal to</td>
</tr>
<tr>
<td>∈</td>
<td>\in</td>
<td>is an element of</td>
</tr>
<tr>
<td>∉</td>
<td>\notin</td>
<td>is not an element of</td>
</tr>
<tr>
<td>⊆</td>
<td>\subseteq</td>
<td>is a subset of</td>
</tr>
<tr>
<td>→</td>
<td>\rightarrow</td>
<td>(various meanings)</td>
</tr>
<tr>
<td>δ</td>
<td>\delta</td>
<td>transition function</td>
</tr>
<tr>
<td>α</td>
<td>\alpha</td>
<td>regular expression</td>
</tr>
<tr>
<td>*</td>
<td>^\ast</td>
<td>Kleene star</td>
</tr>
</tbody>
</table>

5 Drawing automata

Let’s start off with a simple DFA from Sipser [2012] (Figure 1.6). The formal description of the DFA is:

\[ M_1 = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, \{q_2\}) , \]

where \( \delta \) is given by:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>q_1</td>
<td>q_1</td>
<td>q_2</td>
</tr>
<tr>
<td>q_2</td>
<td>q_3</td>
<td>q_2</td>
</tr>
<tr>
<td>q_3</td>
<td>q_2</td>
<td>q_2</td>
</tr>
</tbody>
</table>

Below is the code that generates the state diagram of \( M_1 \).
Below, we’ll go through this example piece by piece.

### 5.1 The tikzpicture environment

Inside the document, each TikZ diagram must reside in a `tikzpicture` environment:

\begin{tikzpicture}
% tikz code goes here
\end{tikzpicture}

### 5.2 Nodes

Let’s start off by drawing the nodes. Nodes can be positioned either manually or relative to other nodes. Relative placement is often much easier.

\begin{tikzpicture}
\node[state, initial] (q1) {$q_1$};
\node[state, accepting, right of=q1] (q2) {$q_2$};
\node[state, right of=q2] (q3) {$q_3$};
\draw (q1) edge[loop above] node {0} (q1);
\draw (q1) edge node {1} (q2);
\draw (q2) edge[loop above] node {1} (q2);
\draw (q2) edge[bend left] node {0} (q3);
\draw (q3) edge[bend left] node {0, 1} (q2);
\end{tikzpicture}

The general form of the `\node` command is:

\node[<options>] (<name>) at (<x>,<y>) {<label>};

The `<options>`, `<name>`, and at `<x>,<y>` are all optional, but the `<label>` is required.

**Options** The options (for finite automata) are:

- `state`: always give this option to draw a node as a state
- `initial`: specifies the start state
- `accepting`: specifies an accept state

Note that the size of a node depends on the length of its label; to force a minimum size (say, 1 inch), use `minimum size=1in`.

**Name** The name of a node is the name by which you refer to the node, when positioning other nodes relative to it or when drawing edges into or out of it.
**Position**  You specify the absolute position of a node using \texttt{at (<x>,<y>)} where \texttt{<x>} and \texttt{<y>} are coordinates (\texttt{<x>} coordinates go to the right; \texttt{<y>} coordinates go up).

Or you can specify a relative position using \texttt{left of=<name>, right of=<name>, above of=<name>, below of=<name>}. There’s also \texttt{above left of=<name>}, etc.\textsuperscript{3}

The \texttt{positioning} library which we have already imported provides some further options.

- \texttt{xshift=x, yshift=y}: Gives manual control of the node positions after relative placement. Eg:

\begin{verbatim}
\node[state, right of=q1, xshift=1cm] (q2) {$q_2$};
\end{verbatim}

**Label**  This can be anything you want. Typically you will surround it with dollar signs to use math mode.

### 5.3 Edges

Once the states are all in place, let’s start adding the transitions, that is, the edges between the states.

The \texttt{draw} command can be used to draw the edges between the already created nodes (states).

\begin{verbatim}
\begin{tikzpicture}
\node[state, initial] (q1) {$q_1$};
\node[state, accepting, right of=q1] (q2) {$q_2$};
\node[state, right of=q2] (q3) {$q_3$};
\draw (q1) edge node {\tt 1} (q2);
\end{tikzpicture}
\end{verbatim}

The general syntax is as follows:

\begin{verbatim}
\draw (<source node>) edge[<options>] node {<label>} (<dest node>);
\end{verbatim}

**Source and destination nodes**  Note that \texttt{<source node>} and \texttt{<dest node>} are the names of the nodes, not their labels.

**Options**  The \texttt{<options>} modify the appearance of the edge.

- For edges that start and end in the same node (self-loops), you must \texttt{loop above, loop above, loop left, or loop right}.
- By default, the edges are straight, so to prevent overlaps use \texttt{bend left} or \texttt{bend right}.
- To modify the placement of the edge label, use \texttt{above} or \texttt{below}.

**Label**  This can be anything you want. Note that Sipser uses typewriter font for symbols, so you probably want to write \texttt{\{\tt 0\}} or \texttt{\texttt{0}}.

**Shorthand**  Multiple edges can be drawn with the same \texttt{draw} command, like so:

\begin{verbatim}
\draw (q1) edge[loop above] node {\tt 0} (q1)
    edge node {\tt 1} (q2);
\draw (q2) edge[loop above] node {\tt 1} (q2)
    edge[bend left] node {\tt 0} (q3);
\draw (q3) edge[bend left] node {\tt 0}, {\tt 1} (q2);
\end{verbatim}

\textsuperscript{3}Technically, these options are deprecated, but we find them useful anyway. See \url{https://tex.stackexchange.com/questions/9386/difference-between-right-of-and-right-of-in-pgf-tikz}.
6 More examples

As another example, let’s draw a NFA (Sipser, Figure 1.42).

\[ N_1 = (\{1, 2, 3\}, \{a, b\}, \delta, 1, \{1\}), \]

where \( \delta \) is given by:

\[
\begin{array}{c|ccc}
   & a & b & \epsilon \\
1 & \{\} & \{2\} & \{3\} \\
2 & \{2, 3\} & \{3\} & \{\} \\
3 & \{1\} & \{\} & \{\} \\
\end{array}
\]

Below is the code that generates the state diagram of \( N_1 \):

\begin{tikzpicture}
  \node[state, initial, accepting] (1) at (1.5,2.6) {$1$};
  \node[state] (2) at (0,0) {$2$};
  \node[state] (3) at (3,0) {$3$};

  \draw (1) edge node{\tt b} (2)
        edge[bend left=15] node {$\epsilon$} (3)
  (2) edge[loop left] node{\tt a} (2)
        edge node{{\tt a}, \tt b} (3)
  (3) edge[bend left=15] node{\tt a} (1);
\end{tikzpicture}

Our final example is the state diagram of the DFA equivalent to the NFA \( N_1 \):

\[ D_2 = (\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}), \{a, b\}, \delta, \{1, 3\}, \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}\}), \]

where \( \delta \) is given by

\[
\begin{array}{c|ccc}
   & a & b & \emptyset \\
\emptyset & \emptyset & \emptyset & \emptyset \\
\{1\} & \emptyset & \{2\} & \emptyset \\
\{2\} & \{2, 3\} & \{3\} & \emptyset \\
\{3\} & \{1, 3\} & \emptyset & \emptyset \\
\{1, 2\} & \{2, 3\} & \{2, 3\} & \emptyset \\
\{1, 3\} & \{1, 3\} & \{2\} & \emptyset \\
\{2, 3\} & \{1, 2, 3\} & \{3\} & \emptyset \\
\{1, 2, 3\} & \{1, 2, 3\} & \{2, 3\} & \emptyset \\
\end{array}
\]

Below is the code that generates the state diagram.
\begin{tikzpicture}
  \tikzset{every state/.append style={rectangle, rounded corners}}
  \node[state, accepting, right of=emp] (1) {$\{1\}$};
  \node[state, right of=1] (2) {$\{2\}$};
  \node[state, accepting, right of=2] (12) {$\{1, 2\}$};
  \node[state, below of=emp] (3) {$\{3\}$};
  \node[state, initial, initial where=above, accepting, right of=3] (13) {$\{1, 3\}$};
  \node[state, right of=13] (23) {$\{2, 3\}$};
  \node[state, accepting, right of=23] (123) {$\{1, 2, 3\}$};

  \draw (emp) edge[loop left] node {{\tt a}, {{\tt b}}} (emp)
  (1) edge[above] node {{\tt a}} (emp)
  (1) edge node {{\tt b}} (2)
  (2) edge node {{\tt a}} (23)
  (2) edge[above] node {{\tt b}} (3)
  (12) edge[auto=right,near start] node {{\tt a}, {\tt b}} (23)
  (3) edge node {{\tt b}} (emp)
  (3) edge node {{\tt a}} (13)
  (13) edge[loop right] node {{\tt a}} (13)
  (13) edge node {{\tt b}} (2)
  (23) edge[bend left,above] node {{\tt a}} (123)
  (23) edge[bend left] node {{\tt b}} (3)
  (123) edge[loop above] node {{\tt a}} (123)
  (123) edge[bend left,below] node {{\tt b}} (23);
\end{tikzpicture}

References
