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. TikZ 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 Overleaf (overleaf.com). If you sign up using your nd.edu address, you’ll get unlimited private projects. For further information regarding setup, visit http://www.latex-project.org/get/.

3 The preamble

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

```latex
\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 this if you need to.
  every state/.append style={ % Sets the style of each state
    semithick,
    fill=gray!10 % Shade nodes
  },
  initial/.append style={
    initial text={}, % No label on start arrow
  }
}
```

1 The Comprehensive TeX Archive Network (CTAN) is the central place for all kinds of material around TeX. https://www.ctan.org/?lang=en

2 https://www.ctan.org/pkg/pgf?lang=en
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>\textvisiblespace</td>
<td>blank symbol</td>
</tr>
<tr>
<td>{ }</td>
<td>{ }</td>
<td>delimiters for sets</td>
</tr>
<tr>
<td>\emptyset</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>\notin</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>^*</td>
<td>Kleene star</td>
</tr>
</tbody>
</table>

5 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:
Below is the code that generates the state diagram of $M_1$.

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

Below, we’ll go through this example piece by piece.

5.1 The \texttt{tikzpicture} environment

Inside the document, each TikZ diagram must reside in a \texttt{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}
% tikz code goes here
\end{tikzpicture}

The general form of the \texttt{node} command is:

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

The \texttt{<options>}, \texttt{(name)}, and at \texttt{(<x>,<y>) are all optional, but the \texttt{<label>} is required.

\textbf{Options} The options (for finite automata) are:

- \texttt{state}: always give this option to draw a node as a state
- \texttt{initial}: specifies the start state
- \texttt{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 \texttt{minimum size=1in}.

\textbf{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 \langle x \rangle, \langle y \rangle} where \langle x \rangle and \langle y \rangle are coordinates (\langle x \rangle coordinates go to the right; \langle y \rangle coordinates go up).

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

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

\begin{itemize}
  \item \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}
\end{itemize}

**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 {$1$} (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.

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

**Label**  This can be anything you want. Note that Sipser uses typewriter font for symbols, so you probably want to write \texttt{\sym{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 {\sym{0}} (q1)
    edge node {\sym{1}} (q2);
\draw (q2) edge[loop above] node {\sym{1}} (q2)
    edge[bend left] node {\sym{0}} (q3);
\draw (q3) edge[bend left] node {\sym{0}, \sym{1}} (q2);
\end{verbatim}

\textsuperscript{3}Technically, these options are deprecated, but we find them useful anyway. See 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:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{}</td>
<td>{2}</td>
<td>{3}</td>
</tr>
<tr>
<td>2</td>
<td>{2,3}</td>
<td>{3}</td>
<td>{}</td>
</tr>
<tr>
<td>3</td>
<td>{1}</td>
<td>{}</td>
<td>{}</td>
</tr>
</tbody>
</table>

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 {$a$} (2) edge[bend left=15] node {$b$} (3);
\draw (2) edge[loop left] node {$a$} (2) edge[below] node {$a, b$} (3);
\draw (3) edge[bend left=15] node {$a$} (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:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
<tr>
<td>{1}</td>
<td>\emptyset</td>
<td>{2}</td>
</tr>
<tr>
<td>{2}</td>
<td>{2,3}</td>
<td>{3}</td>
</tr>
<tr>
<td>{3}</td>
<td>{1,3}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>{1,2}</td>
<td>{2,3}</td>
<td>{2,3}</td>
</tr>
<tr>
<td>{1,3}</td>
<td>{1,3}</td>
<td>{2}</td>
</tr>
<tr>
<td>{2,3}</td>
<td>{1,2,3}</td>
<td>{3}</td>
</tr>
<tr>
<td>{1,2,3}</td>
<td>{1,2,3}</td>
<td>{2,3}</td>
</tr>
</tbody>
</table>

Below is the code that generates the state diagram.

\begin{tikzpicture}
\node[state, initial] (1) at (1.5,2.6) {$\emptyset$};
\node[state] (2) at (2.5,1) {$\{1\}$};
\node[state] (3) at (2,0) {$\{2\}$};
\node[state] (4) at (1,0) {$\{3\}$};
\draw (1) edge node {$a$} (2) edge node {$b$} (4);
\draw (2) edge node {$b$} (3) edge node {$a$} (4);
\draw (3) edge node {$a$} (4);
\end{tikzpicture}
\begin{tikzpicture}
\tikzset{every state/.append style={rectangle, rounded corners}}
\node[state] (emp) {$\emptyset$};
\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 {$\text{a}, \text{b}$} (emp)
\(\rightarrow\)
(1) edge node {$\text{a}$} (emp)
(2) edge node {$\text{b}$} (2)
(2) edge node {$\text{a}$} (23)
(2) edge node {$\text{b}$} (3)
(12) edge (23)
(3) edge node {$\text{b}$} (emp)
(3) edge node {$\text{a}$} (13)
(13) edge (emp)
(13) edge node {$\text{a}$} (13)
(13) edge node {$\text{b}$} (2)
(23) edge node {$\text{b}$} (123)
(23) edge node {$\text{a}$} (3)
(123) edge node {$\text{a}$} (123)
(123) edge node {$\text{b}$} (23);
\end{tikzpicture}

References
