

Graphics in L^AT_EX

Hadi Safari



University of Tehran

ACM Student Chapter

L^AT_EX Intro

Summer 1398

<https://hadisafari.ir/courses/latex>



Floats

1 Floats

2 Simple Graphics

3 More about Graphics

4 PGF/TikZ



Floats

- some parts of document in \LaTeX are *float*, e.g. `figure` & `table` environments
- \LaTeX tries to put floats in a proper position automatically, usually at the top or bottom of the page, instead of putting them in the middle of texts
- the following positioning options are available:
 - `t` top
 - `b` bottom
 - `h` approximately here
 - `p` float-only page
 - `!` override internal parameters \LaTeX uses for determining *good* float positions
 - `H` precisely here requires `float` package
- a combination of options is acceptable, too
 e.g. when using `ht` \LaTeX will try to put the float at the insertion point, then on the top of the next page if it happens to violate its typesetting rules
 consider using `[!htb]` if you want not to use `float` package



Cross-Reference

- labels can be created using `\label{key}` command, usually after caption or section
- `\ref{key}` & `\pageref{key}` can be used to refer to labels
- *keys* traditionally are like *type:name*, when *type* is one of `fig`, `tab`, `chap`, `sec`, `eq`, and `fn`
- `\nameref{key}` from `nameref` package can be used to automatically add prefix to references



Simple Graphics

- 1 Floats
- 2 Simple Graphics
- 3 More about Graphics
- 4 PGF/TikZ



Simple Graphics

- to add external graphics (pictures, images, graphs, etc.) to L^AT_EX document, you need `graphicx` package
- `\includegraphics[options]{file}` command can be used to add graphics, then
- usually, the inserted graphic is put in a `figure` environment, making it float
- it has got an optional argument for positioning
- you may want to use `\centering` declaration at the beginning of figure
- it's possible to add a caption to the float by the `\caption{title}` command
- caption usually should be placed at the bottom of the picture in the scientific documents
- however, you may want to put the caption at the top of picture to make picture visible after clicking a cross-reference to it



Simple Graphics

Example

```
\usepackage{graphicx}
...
\begin{figure}[h]
  \centering
  \includegraphics[width=0.5\textwidth]{ut}
  \caption{Logo of
    \textit{University of Tehran}}
  \label{fig:ut-logo}
\end{figure}
```



Figure: Logo of *University of Tehran*



More about Graphics

- 1 Floats
- 2 Simple Graphics
- 3 **More about Graphics**
- 4 PGF/TikZ



Side Caption

- `sidecap` package can be used to create captions appear at the side of float
- Example:

```
\begin{SCfigure}
  \centering
  \caption{University of Tehran is the
    oldest modern university located
    in Tehran, Iran. It is also one of
    the most prestigious universities
    in the Middle East [Wikipedia].}
  \includegraphics[width=0.3\textwidth]
    {../../img/ut}
\end{SCfigure}

\end{document}
```



Figure 1: University of Tehran is the oldest modern university located in Tehran, Iran. It is also one of the most prestigious universities in the Middle East [Wikipedia].



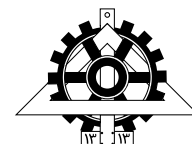
Subfloat

- `subcaption` package can be used to create subfloats within a single float
- Example:

```
\begin{figure}
  \centering
  \begin{subfigure}[b]{0.3\textwidth}
    \centering
    \includegraphics[width=0.8\textwidth]
      {../../img/ut}
    \caption{University of Tehran}
    \label{fig:ut}
  \end{subfigure}
  ~
  \begin{subfigure}[b]{0.3\textwidth}
    \centering
    \includegraphics[width=\textwidth]
      {../../img/fanni}
    \caption{College of Engineering}
    \label{fig:eng}
  \end{subfigure}
\end{figure}
```



(a) University of Tehran



(b) College of Engineering

Figure 1: UT Engineering

- `subfig` package can be used, too



Wrapping Text around Figures

- `wrapfig` package can be used to allow text to wrap around a float
- `\begin{wrapfigure}[lineheight]{position}[overhang]{width}` environment provides the aforementioned functionality
- Example:

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \centering
  \includegraphics[width=0.48\textwidth]
    {../../img/ut}
  \caption{University of Tehran}
\end{wrapfigure}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.



Figure 1: University of Tehran



Wide Figures in Two-column Documents

- the starred version of `figure` environment can be used to create floats as wide as the page in the two-sided documents



PGF/TikZ

- 1 Floats
- 2 Simple Graphics
- 3 More about Graphics
- 4 PGF/TikZ



Procedural Graphics

- `picture` environment
 - pure L^AT_EX
 - very limited
 - `epic` & `eepic`
- `pstricks` environment
 - uses PostScript possibilities
 - won't work with pdfL^AT_EX, etc.
- MetaPost & METAFONT
- PGF/TikZ
- see https://en.wikibooks.org/wiki/LaTeX/Introducing_Procedural_Graphics



PGF/TikZ

- a pair of languages for producing vector graphics from a geometric/algebraic description
- ‘Portable Graphics Format’ & ‘TikZ ist *kein* Zeichenprogramm’ (German for ‘TikZ is *not* a drawing program’)
- both PDF and PostScript
- While the previous systems (`picture`, `epic`, `pstricks` or `metapost`) focus on the *how* to draw, `TikZ` focuses more on the *what* to draw



PGF/TikZ in L^AT_EX

- `tikz` package should be used to create PGF/TikZ graphics
- `\usetikzlibrary{commaseparatedlistoflibraries}` should be used to add extra libraries, e.g. `arrows`, `trees`, `automata`, `mindmap`, `petri`, `shapes.geometric`, etc.
- you may need to include extra packages, too
- `\begin{tikzpicture}[options] cnt\end{tikzpicture}` OR `\tikz[options]{cnt}` commands can be used to create PGF/TikZ graphics



PGF/TikZ in L^AT_EX

Cont.

- each line of TikZ ends with ;
- `\draw` command can be used to draw different shapes
- `\node[options](name)at(coord){text};` &
`\coordinate[options](name)at(coord);` can be used to create *nodes*
- `\path` command is a shortcut for creating nodes and drawing line between them
- see <https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>



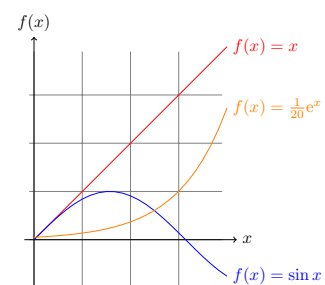
PGF/TikZ Example

Plotting Functions

```

\begin{tikzpicture}[domain=0:4]
  \draw[very thin,color=gray] (-0.1,-1.1) grid
    (3.9,3.9);
  \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
  \draw[color=red] plot (\x,\x) node[right]
    {$f(x) = x$};
  \draw[color=blue] plot (\x,{sin(\x r)}) node[right]
    {$f(x) = \sin x$};
  \draw[color=orange] plot (\x,{0.05*exp(\x)})
    node[right] {$f(x) = \frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}

```



PGF/TikZ Example

Tree

```

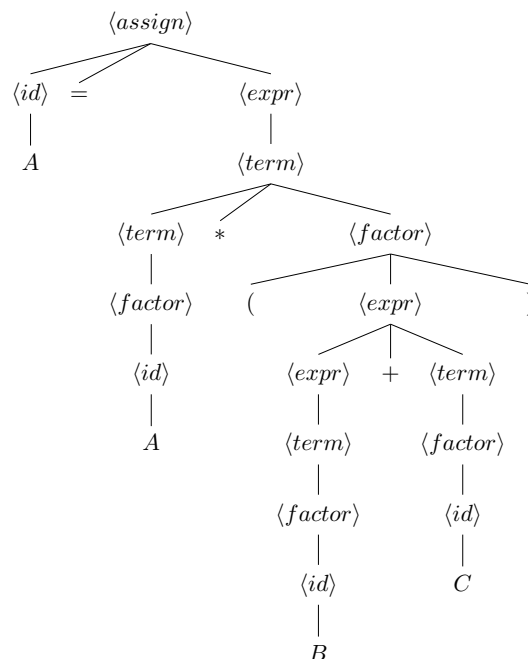
\usepackage{tikz-qtree}
\begin{tikzpicture}
\Tree
  [.\langle assign\rangle$
    [.\langle id\rangle$
      $A$
    ]
    $=$
    [.\langle expr\rangle$
      [.\langle term\rangle$
        [.\langle term\rangle$
          [.\langle factor\rangle$
            [.\langle id\rangle$
              $A$
            ]
          ]
        ]
      ]
      $*$
      [.\langle factor\rangle$
        (
          [.\langle expr\rangle$
            [.\langle expr\rangle$
              [.\langle term\rangle$
                [.\langle factor\rangle$
                  [.\langle id\rangle$
                    $A$
                  ]
                ]
              ]
              $+$
              [.\langle term\rangle$
                [.\langle factor\rangle$
                  [.\langle id\rangle$
                    $B$
                  ]
                ]
              ]
            ]
            $)$
          ]
          [.\langle factor\rangle$
            [.\langle id\rangle$
              $C$
            ]
          ]
        )
      ]
    ]
  ]
\end{tikzpicture}

```



PGF/TikZ Example

Tree: Cont.

Figure 1: parse tree of $A = A * (B + C)$ 

PGF/TikZ Example

Automata

```

\usetikzlibrary{arrows,automata,positioning}
\begin{tikzpicture}[>,>stealth',auto]
\node[state] (s04) {\langle s_0, s_4 \rangle};
\node[state, accepting] (s14) [below = of s04]
{\langle s_1, s_4 \rangle};
\node[state] (s01) [above left = of s04] {\langle s_0, s_1 \rangle};
\node[state, accepting] (s11) [below left = of s01]
{\langle s_1, s_1 \rangle};
\node[initial, state] (s0i) [left = of s01]
{\langle s_0, s_i \rangle};
\node[state, accepting] (s1i) [below left = of s11]
{\langle s_1, s_i \rangle};
\node[state] (s02) [above right = of s04] {\langle s_0, s_2 \rangle};
\node[state, accepting] (s12) [below right = of s02]
{\langle s_1, s_2 \rangle};
\node[state, accepting] (s13) [below left = of s14]
{\langle s_1, s_3 \rangle};
\node[state] (s03) [above left = of s13] {\langle s_0, s_3 \rangle};
\node[state, accepting] (s15) [below right = of s14]
{\langle s_1, s_5 \rangle};
\node[state] (s05) [above right = of s15] {\langle s_0, s_5 \rangle};
\draw (s0i) to[bend left] node {\langle a \rangle} (s01);
\draw (s0i) to[bend right] node[left] {\langle a \rangle}
(s11);
\draw (s01) to[bend left] node[above] {\langle b \rangle}
(s04);
\draw (s01) to node[above] {\langle b \rangle} (s14);
\draw (s11) to node[above] {\langle b \rangle} (s14);
\draw (s02) to node[below] {\langle b \rangle} (s04);
\draw (s02) to node[below] {\langle b \rangle} (s14);
\draw (s12) to node[below] {\langle b \rangle} (s14);
\draw (s04) to[bend left] node[above] {\langle c \rangle}
(s02);
\draw (s03) to[bend left] node[above] {\langle b \rangle}
(s04);
\draw (s03) to[bend right] node[below] {\langle b \rangle}
(s14);
\draw (s13) to[bend right] node[below] {\langle b \rangle}
(s14);
\draw (s04) to[bend left] node[below] {\langle b, c \rangle}
(s03);
\draw (s05) to[bend left] node[below] {\langle b \rangle}
(s04);
\draw (s05) to[bend left] node[below] {\langle b \rangle}
(s14);
\draw (s04) to[bend left] node[above] {\langle a, b,
c \rangle} (s05);
\draw (s05) to[loop below] node {\langle a, b, c \rangle}
(s05);
\end{tikzpicture}

```

PGF/TikZ Example

Automata: Cont.

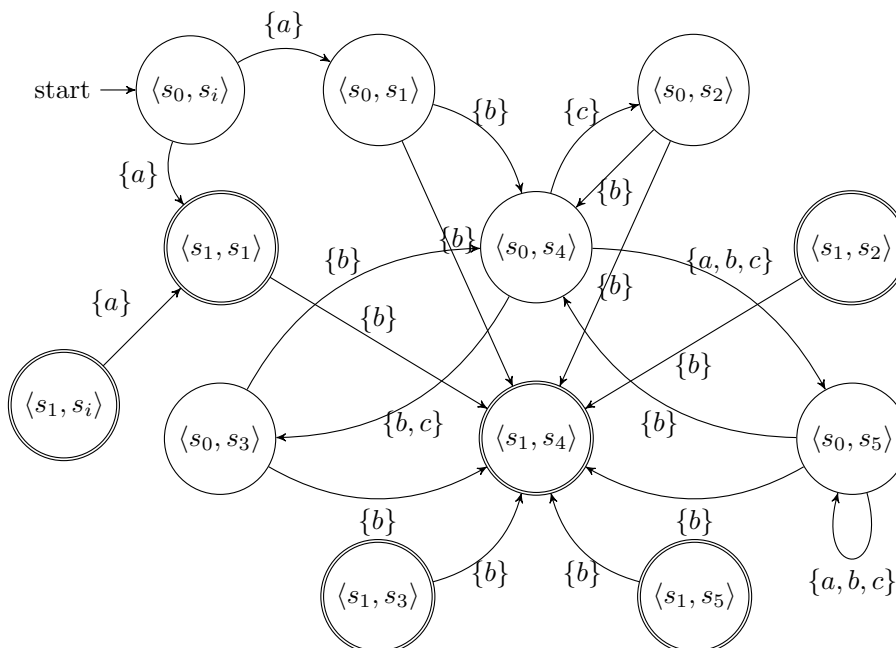


Figure 1: $A_M \times A_{-\phi}$

PGF/TikZ Example

Timing

```

\usepackage{tikz-timing}
\begin{figure}
\centering
\caption{Timing diagram of
74LS191\label{fig:74LS191_timing}}
\begin{tikztimingtable}
\ol{load} & & 2H N(Lst) 1.5L N(Let) 24.5H \\
$A$ & & 1L 3H 24U \\
$B$ & & 1L 3L 24U \\
$C$ & & 1L 3H 24U \\
$D$ & & 1L 3H 24U \\
$DCBA$ & & 1D{0} 3D{13} 24U \\
down/\ol{up} & & 1H 13.5L 13.5H \\
\ol{enable} & & 1H 11.5L N(Ist) 4HN (Iet)
11.5L \\
clock & & [C] 4{C} N(Ust) 14{C} N(Dst)
10{C} \\
$Q_A$ & & UU HH LL HH LL HH LL LL LL HH
LL HH LL HH \\
$Q_B$ & & UU LL HH HH LL LL HH HH HH LL
LL HH HH LL \\
$Q_C$ & & UU HH HH HH LL LL LL LL LL LL
LL HH HH HH \\
$Q_D$ & & UU HH HH HH LL LL LL LL LL LL
LL HH HH HH \\
$Q_DQ_CQ_BQ_A$ & & UU 2D{13} 2D{14} 2D{15} 2D{0}
2D{1} 2D{2} 2D{2} 2D{2} 2D{1} 2D{0}
2D{15} 2D{14} 2D{13} \\
max/\ol{min} & & UU LL LL HH LL LL LL LL LL
HH LL LL LL \\
\ol{ripple clock} & & UU HH HH HL HH HH HH HH
HH HL HH HH HH \\
& & 2S N(Ls) 1.5S N(Le) 0.5S N(Us) 8.5S N(Ue)
N(Is) 4S N(Ie) 1.5S N(Ds) 10S N(De) \\
\extracode
\draw[<->] (Ls) -- (Le) node[midway, below]
{load};
\draw[<->] (Us) -- (Ue) node[midway, below]
{count up};
\draw[<->] (Is) -- (Ie) node[midway, below]
{inhibit};
\draw[<->] (Ds) -- (De) node[midway, below]
{count down};
\draw[help lines] (Ls) -- (Lst);
\draw[help lines] (Le) -- (Let);
\draw[help lines] (Is) -- (Ist);
\draw[help lines] (Ie) -- (Iet);
\draw[help lines] (Us) -- (Ust);
\draw[help lines] (Ds) -- (Dst);
\end{tikztimingtable}

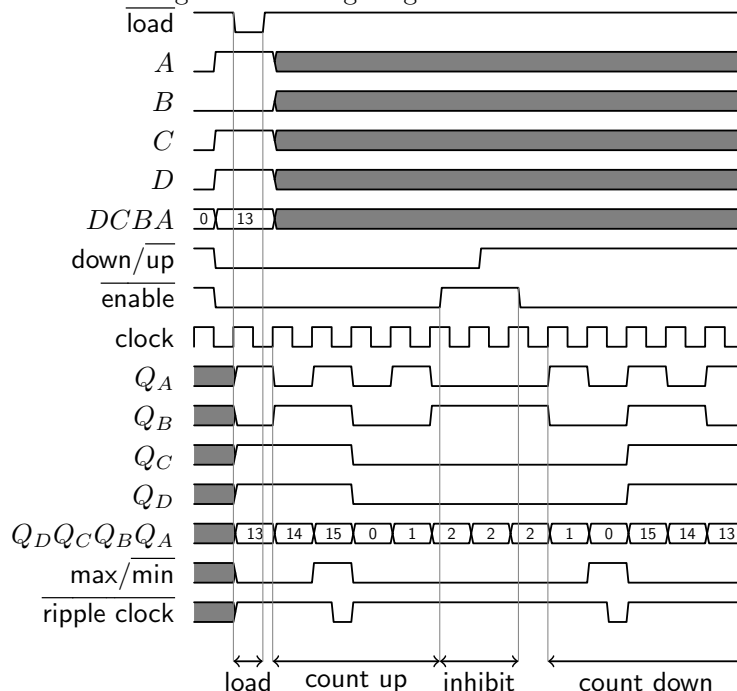
```



PGF/TikZ Example

Timing: Cont.

Figure 1: Timing diagram of 74LS191



PGF/TikZ Example

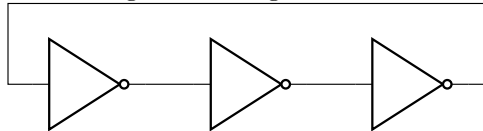
Circuit

```

\usepackage{circuitikz}
\begin{circuitikz}
  \draw
    (1,0) node (nand1) [not port] {}
    (3,0) node (nand2) [not port] {}
    (5,0) node (nand3) [not port] {}
  ;
  \draw (nand1.out) -- (nand2.in);
  \draw (nand2.out) -- (nand3.in);
  \draw (nand3.out) -- (6,0) -- (6,1) -- (0,1) -- (0,0) -- (nand1.in);
\end{circuitikz}

```

Figure 1: Ring oscillator



PGF/TikZ Example

UML Sequence Diagram

```

\usepackage{pgf-umlsd}
\begin{figure}
\centering
\begin{sequencediagram}
\newthread{cli}{:Client}
\newinst[3]{cas}{:CASServer}
\newinst[3]{app}{:WebApp}
\begin{call}{cli}{login()}{app}{Hi Hadi!}
  \begin{call}{app}{getTicket(casLoginUrl)}{cli}{ticket}
    \begin{call}{cli}{login(username, password)\postlevel}{cas}{ticket}
      \begin{call}{cas}{\shortstack{checkCredentials\\(username, password)}}{cas}{True}
        \end{call}
      \begin{call}{cas}{getTicket(username)}{cas}{ticket}
        \end{call}
      \end{call}
    \end{call}
  \end{call}
  \begin{call}{app}{validate(ticket)}{cas}{\texttt{"name": "Hadi"}}
    \begin{call}{cas}{validate(ticket)}{cas}{username, \dots}
      \end{call}
    \begin{call}{cas}{getName(username)}{cas}{Hadi}
      \end{call}
    \end{call}
  \end{call}
\end{sequencediagram}
\caption{Central Authentication System (CAS) work flow\label{fig:cas-sequence-diagram}}
\end{figure}

```



PGF/TikZ Example

UML Sequence Diagram: Cont.

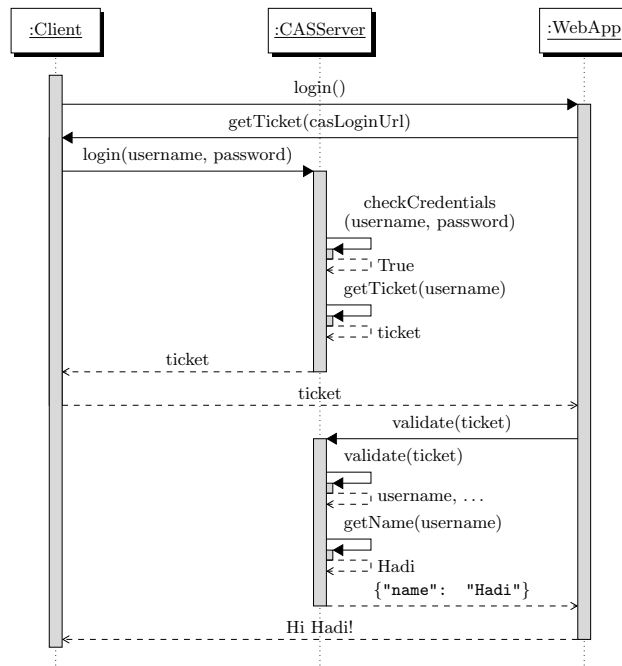


Figure 1: Central Authentication System (CAS) work flow



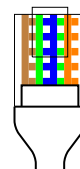
PGF/TikZ Example

T_{568B}





```

\begin{tikzpicture}
  \foreach \c/\x/\s in {brown/0, brown/1/1, green/2,
    blue/3/1, blue/4, green/5/1, orange/6, orange/7/1} {
    \draw[line width=0.1cm, color=\c, \if\s1dashed\fi] (\x *
      \the\pgflinewidth, 0) -- (\x * \the\pgflinewidth,
      1);
  }
  \draw[] (-0.15, -0.3) -- (-0.15, 1) -- (0.85, 1) -- (0.85,
    -0.3);
  \draw[thick] (-0.15, -0.3) -- (0.85, -0.3);
  \draw[thick, rounded corners=1mm] (0.15, -1.2) -- (0.15,
    -1) -- (-0.15, -0.6) -- (-0.15, -0.3) (0.85, -0.3) --
    (0.85, -0.6) -- (0.55, -1) -- (0.55, -1.2);
  \draw[thick] (-0.05, -.3) -- (-0.05, 0) -- (0.75, 0) --
    (0.75, -.3);
  \draw[] (0.1, 0.4) -- (0.1, 1.1) -- (0.6, 1.1) -- (0.6,
    0.4) -- (0.1, 0.4);
\end{tikzpicture}

```

Figure 1: T_{568B}

References

-  Woerner, K., Benjamin M., and Henrik S. \LaTeX Lab. Department of Mechanical Engineering of MIT. 2017. Available Online at http://oceanai.mit.edu/k_w/latex_lab/.
-  Higginbottom, R. Introduction to Scientific Typesetting. 2012. Available Online at <http://www2.washjeff.edu/users/rhigginbottom/latex/main.html>.
-  Wikibooks contributors. \LaTeX . Wikibooks, The Free Textbook Project. Available Online at <https://en.wikibooks.org/wiki/LaTeX>.
-  Oetiker T, Partl H., and Hyna I. Not So Short Introduction to \LaTeX 2 ϵ . Available Online at <http://mirrors.ctan.org/info/lshort/english/lshort.pdf>.

