The **string-diagrams** package*

Draw string diagrams using TikZ

Paolo Brasolin

paolo.brasolin@gmail.com

v0.2.1 (2023/06/13)

Please note this is the major version zero, meant for initial development: *anything MAY change at any time*. The upside is that this is the best time to **contribute**! Of course you can also just keep the sty along with your code and not care at all.

1 Documentation

To draw boxes, you use this style on a node.

\begin{tikzpicture}
\node[box] {A};
\end{tikzpicture}

A

You can draw multiple boxes using any of your standard TikZ positioning techniques. Don’t forget to label the nodes so you can easily reference them.

\begin{tikzpicture}
\node[box] (A) at (0,0) {A};
\node[box, right of=A] (B) {B};
\node[box] (C) at ($(B)+(2cm,1em)$) {C};
\end{tikzpicture}

A  B  C

*Thanks!
You can open up any number of ports on any side of a box using the appropriate key. Then, you can refer to the opened ports by their index.

\begin{tikzpicture}
  \node [box, box ports north=3, box ports east=3, box ports south=3, box ports west=3, minimum width=6em, minimum height=6em] (A) {A};
  \foreach \side in {north, east, south, west} \foreach \index in {1,...,3} \node [marker] at (A.\side.\index) {\index};
\end{tikzpicture}

The box ports key is a shortcut to set the number of ports on all sides at once.

\begin{tikzpicture}
  \node [box, box ports=1/2/3/4] (A) {A};
  \foreach \side/\n in {north/1, east/2, south/3, west/4} \foreach \index in {1,...,\n} \node [marker] at (A.\side.\index) {};
\end{tikzpicture}

The same value can also be passed to the box key itself.
To connect boxes, you can use the \texttt{wires} macro. The first argument is TikZ styling for the wires; the second argument is a nested dictionary specifying the connectivity; the third argument is a list of the loose ends to draw. Boxes have the following anchors: \texttt{west}, \texttt{west.0}, \texttt{west.1}, \texttt{east}, \texttt{east.0}, and \texttt{east.1}.

\begin{tikzpicture}[scale=0.6]
  \node[box=0/2/0/1] (A) at (-2, 0) {A};
  \node[box=0/1/0/2] (B) at (+2, 0) {B};
  \node[box=0/1/0/1] (C) at ( 0,+1) {C};
  \node[box=0/1/0/1] (D) at ( 0,-1) {D};
  \wires{
    A = { east.1 = C.west, east.2 = D.west },
    C = { east = B.west.1 },
    D = { east = B.west.2 },
  }{ A.west, B.east }
\end{tikzpicture}

To split and join wires, you can use dots and their anchors \texttt{north}, \texttt{east}, \texttt{south}, and \texttt{west}. Remember to have fun with styling wires.

\begin{tikzpicture}
  \node[box=0/1/0/2] (A) at ( 0,+1) {A};
  \node[box=0/2/0/1] (B) at ( 0,-1) {B};
  \node[dot] (x) at (+1, 0) {};
  \node[dot] (y) at (-1, 0) {};
  \wires[looseness=1.5, dashed]{
    A = { east = x.north },
    B = { east.1 = x.south },
    y = { north = A.west.2, south = B.west },
  }{ A.west.1, B.east.2, x.east, y.west }
\end{tikzpicture}

That’s it. This is the package, for now.

## 2 Implementation

Open the DocStrip guards and set the internal namespace prefix (as per \LaTeX{} DocStrip convention).

\begin{verbatim}
(\*package)
(\@\*=\stridi)
\end{verbatim}

Load the essential support (\texttt{expl3}) “up-front”.

\begin{verbatim}
\RequirePackage{expl3}[2023/05/11]
\RequirePackage{tikz}[2023/01/15]
\end{verbatim}
Identify the package and give the overall version information.

\ProvidesExplPackage{string-diagrams}{2023/06/13}{0.2.1}{Draw string diagrams using TikZ}

Define high level keys to configure the number of ports on each side.

\begin{verbatim}
\pgfkeys{
  /pgf/box-ports-north/.initial=1,
  /pgf/box-ports-east/.initial=1,
  /pgf/box-ports-south/.initial=1,
  /pgf/box-ports-west/.initial=1,
  /pgf/box-ports/.style-args={#1/#2/#3/#4}{
    /pgf/box-ports-north=#1,
    /pgf/box-ports-east=#2,
    /pgf/box-ports-south=#3,
    /pgf/box-ports-west=#4,
  },
\}
\end{verbatim}

(End of definition for /pgf/box ports north and others. These functions are documented on page 2.)

\begin{verbatim}
\__stridi_intersect_hv_lines_through:NN\end{verbatim}

Calculates the intersection of two lines parallel to axes passing through given points on the plane.

#1: Point through which the vertical line passes
#2: Point through which the horizontal line passes

\begin{verbatim}
\cs_new:Nn \__stridi_intersect_hv_lines_through:NN {
  \pgfextractx { \pgf@xa } { #1 }
  \pgfextracty { \pgf@ya } { #2 }
  \pgfpoint { \pgf@xa } { \pgf@ya }
}\end{verbatim}

(End of definition for \__stridi_intersect_hv_lines_through:NN.)

\begin{verbatim}
\__stridi_subdivide_segment:nNNNNN\end{verbatim}

Defines macros numbering equally spaced points on a segment.

#1: Base namespace
#2: Points count
#3: Point containing the x coordinate of the starting point
#4: Point containing the y coordinate of the starting point
#5: Point containing the x coordinate of the ending point
#6: Point containing the y coordinate of the ending point

\begin{verbatim}
\cs_new:Nn \__stridi_subdivide_segment:nNNNNN {
  \int_step_inline:nnnn { #2 } { -1 } { 1 } { #1 .##1 }{ \prg_break: { \prg_do_nothing: {}}}
}\end{verbatim}
box  Define a rectangular shape with configurable ports.

\pgfdeclare{box}{
\% Inherit all the structure of rectangle
\inherit{rectangle}{north~west, north, north~east,
west, center, east,
mid~west, mid, mid~east,
base~west, base, base~east,
south~west, south, south~east,
}
\inherit{rectangle}{#1}
\inherit{rectangle}{#1}
\% Dump port counts into saved macros
\savedmacro{portsnorth}{\pgfmath{\pgfkeysvalueof{/pgf/box~ports~north}}}
\savedmacro{portseast}{\pgfmath{\pgfkeysvalueof{/pgf/box~ports~east}}}
\savedmacro{portssouth}{\pgfmath{\pgfkeysvalueof{/pgf/box~ports~south}}}
\savedmacro{portswest}{\pgfmath{\pgfkeysvalueof{/pgf/box~ports~west}}}
\% Add ports definitions to shape definition
\expandafter\pgfutil\g@addto@macro\csname pgf@sh@s@box\endcsname{
\__stridi_subdivide_segment:nNNNNN { pgf@anchor@box@north } { \portsnorth } { \southwest } { \northeast } { \northeast }
\__stridi_subdivide_segment:nNNNNN { pgf@anchor@box@east } { \portseast } { \northeast } { \northeast } { \southwest }
\__stridi_subdivide_segment:nNNNNN { pgf@anchor@box@south } { \portssouth } { \southwest } { \northeast } { \southwest }
\__stridi_subdivide_segment:nNNNNN { pgf@anchor@box@west } { \portswest } { \southwest } { \southwest } { \southwest }
\begin{verbatim}
\ExplSyntaxOff
\tikzset{  
  box/.default={0/0/0/0},  
  box/.style args={#1}{  
    shape=box,  
    draw,  
    inner sep=.5em,  
    minimum width=2em,  
    minimum height=2em,  
    execute at begin node=$,  
    execute at end node=$,  
    /pgf/box ports=#1,  
  },  
}
\ExplSyntaxOn
(End of definition for \textit{/pgf/box}. This function is documented on page 1.)

\textbf{/pgf/dot} Define style to draw dots.
\begin{verbatim}
\ExplSyntaxOff
\tikzset{  
  dot/.style={  
    shape=circle,  
    fill,  
    inner sep=0,  
    minimum width=0.4em,  
  },  
}
\ExplSyntaxOn
(End of definition for \textit{/pgf/dot}. This function is documented on page 3.)
\end{verbatim}

`\_stridi_draw_bound_wires:nn` Draws bound wires.

\begin{verbatim}
#1: TikZ keys  
#2: dictionary of port labels
\cs_new:Nn \__stridi_draw_bound_wires:nn {  
  \prop_set_from_keyval:Nn \l_tmpa_prop { #2 }  
  \prop_map_inline:Nn \l_tmpa_prop  
  {    
    \prop_set_from_keyval:Nn \l_tmpb_prop { #2 }  
    \prop_map_inline:Nn \l_tmpb_prop  
    {    
      \regex_match_case:nn  
    }  
  }  
}\end{verbatim}

(End of definition for \textit{\_stridi_draw_bound_wires:nn}. This function is documented on page 6.)
\begin{verbatim}
\__stridi_draw_bound_wires:nn \regex_match_case:nn \draw [out={\tl_use:N \g_tmpb_tl}, in={\tl_use:N \g_tmpa_tl}, #1, ] (##1.####1) to (####2);
\end{verbatim}

\__stridi_draw_loose_wires:nn \cs_new:Nn \__stridi_draw_loose_wires:nn \clist_set:Nn \l_tmpa_clist { #2 } \clist_map_inline:Nn \l_tmpa_clist {
\regex_match_case:nn
\draw [out={\tl_use:N \g_tmpb_tl}, in={\tl_use:N \g_tmpa_tl}, #1, ] (##1.####1) to (####2);
\} \}

\end{verbatim}

\begin{verbatim}
\__stridi_draw_loose_wires:nn \cs_new:Nn \__stridi_draw_loose_wires:nn \clist_set:Nn \l_tmpa_clist { #2 } \clist_map_inline:Nn \l_tmpa_clist {
\regex_match_case:nn
\draw [out=-90, in=0, #1] (##1) -- +(0,+1); % TODO: cleaner solution?
\} \}
\end{verbatim}

\begin{verbatim}
\wires \NewDocumentCommand{\wires}{O{0} m m }\__stridi_draw_bound_wires:nn { #1 } { #2 } \__stridi_draw_loose_wires:nn { #1 } { #3 }
\end{verbatim}

---

\textbf{\_\_stridi\_\_draw\_loose\_\_wires:nn} \begin{itemize}
\item \textbf{\#1}: TikZ keys
\item \textbf{\#2}: list of port labels
\end{itemize}

\textbf{\wires} Define our main actor.
Close the DocStrip guards and call it a day.

Change History

0.1.0
General: initial version . . . . . . . . . . . . 1

0.2.0
General: make box ports configurable . 1
/pgf/box: acts as a shortcut for
setting port counts . . . . . . . . . . . . . . 6

0.2.1
\wires: now correctly handles optional
style parameter . . . . . . . . . . . . . . . . 7

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ . . . 119, 120, 121, 122, 144, 145, 150, 151</td>
<td>\endcsname</td>
</tr>
<tr>
<td>/pgf/box</td>
<td>\expandedafter</td>
</tr>
<tr>
<td>/pgf/box_ports</td>
<td>\ExplSyntaxOff</td>
</tr>
<tr>
<td>/pgf/box_ports_east</td>
<td>\ExplSyntaxOn</td>
</tr>
<tr>
<td>/pgf/box_ports_north</td>
<td>\inheritanchor</td>
</tr>
<tr>
<td>/pgf/box_ports_south</td>
<td>\inheritanchorborder</td>
</tr>
<tr>
<td>/pgf/box_ports_west</td>
<td>\inheritbackgroundpath</td>
</tr>
<tr>
<td>/pgf/dot</td>
<td>\inherititsavedanchors</td>
</tr>
<tr>
<td>\l_tmpa_clist</td>
<td>int commands:</td>
</tr>
<tr>
<td>\clist_map_inline:Nn</td>
<td>\int_step_inline:nnn</td>
</tr>
<tr>
<td>\clist_map_inline:nn</td>
<td></td>
</tr>
<tr>
<td>\clist_set:Nn</td>
<td>\NewDocumentCommand</td>
</tr>
<tr>
<td>\clist_utils:n</td>
<td>\northeast</td>
</tr>
<tr>
<td>\cs_new:Nn</td>
<td>\pgfdecoration</td>
</tr>
<tr>
<td>\cs_new_nopar:Npn</td>
<td>\pgfextractx</td>
</tr>
<tr>
<td>\csname</td>
<td>\pgfextracty</td>
</tr>
<tr>
<td>\cs_if_exist:NTF</td>
<td>\pgfkeys</td>
</tr>
<tr>
<td>\cs_new:Nn</td>
<td>\pgfkeysvalueof</td>
</tr>
<tr>
<td>\cs_set:Nn</td>
<td>\pgfmathdivide</td>
</tr>
<tr>
<td>\cs_set_at:Nn</td>
<td>\pgfmathresult</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>box: make ports configurable through TikZ keys</td>
</tr>
<tr>
<td>box: make ports configurable through Ti\kZ keys</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>box</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>\clist_map_inline:Nn</td>
</tr>
<tr>
<td>\clist_map_inline:nn</td>
</tr>
<tr>
<td>\clist_set:Nn</td>
</tr>
<tr>
<td>\l_tmpa_clist</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>\draw</td>
</tr>
</tbody>
</table>