The memorygraphs package

Camil Staps

Sunday 6th January, 2019

Contents
1 Introduction 1
2 Functionality 1
  2.1 Nodes ........................................ 2
  2.2 Markings .................................... 2
  2.3 Anchors .................................... 3
  2.4 References ................................ 4
3 Examples 4

Index 5

1 Introduction
This is the documentation of the \LaTeX package memorygraphs. It defines some TikZ styles and adds anchors to existing styles that ease the declaration of “memory graphs”. It is intended for graphs that represent the memory of a computer program during its execution.

2 Functionality

\texttt{/tikz/memory graph}

The memory graph style is to be used on \texttt{tikzpicture}. It sets a different node distance that the author finds suitable for this kind of graphs.

\begin{tikzpicture}[memory graph]
  \node [draw] (x) {37};
  \node [draw, right=of x] {42};
\end{tikzpicture}

\*info@camilstaps.nl
2.1 Nodes

The following styles can be used to typeset memory blocks:

/tikz/block

This is the most basic style to define a memory block. By default, this shape is a rectangle with borders:

```
\begin{tikzpicture}[memory graph]
\node[block] {37};
\end{tikzpicture}
```

/tikz/arity = ⟨n⟩

The arity style can be used to create a node with arguments. This implies block:

```
\begin{tikzpicture}[memory graph]
\node[arity=2] {Cons};
\end{tikzpicture}
```

\arg{i}

Because blocks with arity are multipart rectangles, one can use TikZ’s \texttt{nodepart} to put contents in the arguments. However, it can be confusing that \texttt{nodepart{two}} refers to the \texttt{first} argument, so we redefine \texttt{arg} in blocks to identify arguments of the memory block:

```
\begin{tikzpicture}[memory graph]
\node[arity=2,block mark north east] {Cons \arg{1} 37 \arg{2} \dots};
\end{tikzpicture}
```

Should one want to use math mode’s \texttt{arg} in a memory block, they can first rename it:

```
\let\matharg\arg
```

\begin{tikzpicture}[memory graph]
\node[block] {$\matharg(1)$};
\end{tikzpicture}
```

2.2 Markings

It is possible to mark the head of memory blocks using triangles in the north east and south east corners.

/tikz/block mark north east = ⟨style⟩
/tikz/block mark north west = ⟨style⟩
/tikz/block mark south east = ⟨style⟩
/tikz/block mark south west = ⟨style⟩

With this key, triangular marks can be added to the corners of the head of a node:

```
\begin{tikzpicture}[memory graph]
\node[arity=2,block mark north east] {Cons};
\end{tikzpicture}
```

It is optional to add a style:
The key is long to avoid clashes with other packages, and because it depends on the context what nodes should be marked for. It is of course possible to define a shorthand in your own document. One application is to mark nodes that are in head normal form (HNF), for which one may define the key `hnf`:

```latex
\begin{tikzpicture}
[memory graph,every node/.style={block}]
\node[block mark north east={fill,red}] at (1,1) {1};
\node[block mark north west={blue}] at (0,1) {2};
\node[block mark south west={fill,green}] at (0,0) {3};
\node[block mark south east={brown}] at (1,0) {4};
\end{tikzpicture}
```

The size of the rectangles is defined by `\memorygraphs@marklength`, which can of course be changed. The default is 4.0pt.

```latex
\makeatletter\memorygraphs@marklength=7pt\makeatother
\begin{tikzpicture}
[memory graph]
\node[arity=2,block mark north west=fill]
{Cons \arg{1}$\ldots$ \arg{2}$\ldots$};
\end{tikzpicture}
```

### 2.3 Anchors

Because blocks with `arity` are multipart rectangles, one can use anchors like `two south` to refer to the south of the second part of a node. These are aliased as `arg i south` (and similar for other anchors on multipart nodes), where `arg 1` stands for `two`. The first block of a node is aliased as `head` instead of `arg 0`, so one can use `head south`. For `head`, anchors for the corners (`head north east`, etc.) are defined as well.

The parts of multipart rectangles do not normally have a `center` anchor, but `memorygraphs` defines these. One can use both `two center` and `arg 1 center` to refer to the center of the first argument of a node.

The additional anchors are shown below. See the TikZ manual for the predefined anchors.
2.4 References

This is a simple style for arrows with a circle at the start and slightly rounded corners:

\begin{tikzpicture}[memory graph]
\node [arity=2, fill=yellow!30, color=black!30, inner xsep=1.25cm, inner ysep=.75cm, line width=.25cm] (v) {Head \arg{1} 1 \arg{2} 2};
\foreach \anchor/\placement in 
{head north/above, head south/below, head center/below, 
 head north east/below, head south east/above, head north west/left, head south west/left, 
 arg 1 north/above, arg 1 south/below, arg 1 center/below, 
 arg 2 north/above, arg 2 south/below, arg 2 center/below}
\draw[shift=(v.\anchor)] plot[mark=x] coordinates{(0,0)} node[\placement] {\scriptsize\texttt{v.\anchor}};
\end{tikzpicture}

3 Examples

- The linked list of Fibonacci numbers on the title page was generated with:

\begin{tikzpicture}
\node[arity=2] (hd) {Cons};
\node[arity=2, below=of hd.arg 2 center, anchor=head north] (tl) {Cons};
\draw[ref] (hd.arg 2 center) -- (tl.head north);
\draw[ref] (tl.arg 2 center) |- ($(hd.head north)+(0,.4)$) -- (hd.head north);
\end{tikzpicture}
A cyclical linked list, with unboxed integers:

\begin{tikzpicture}[memory graph]
\node[block,arity=2] (xs) {Cons \arg{1} 1};
\node[block,arity=2,right=of xs.arg 2 east] (xsb) {Cons \arg{1} 2};
\draw[ref] (xs.arg 2 center) -- (xsb);
\node[block,arity=2,right=of xsb.arg 2 east] (xsc) {Cons \arg{1} 3};
\draw[ref] (xsb.arg 2 center) -- (xsc);
\draw[ref] (xsc.arg 2 center) -- +(0,.6) -| (xs.head north);
\end{tikzpicture}

Index

anchors, 3
[arg, 2
arity key, 2
block key, 2
block mark north east key, 2
block mark north west key, 2
block mark south east key, 2
block mark south west key, 2
memory graph key, 1
ref key, 4