TikZiT Quantum Template

October 23, 2021

The standard way to use files produced by TikZiT is to place the `.tikz` files in the `figures` subdirectory, and include them via the `\tikzfig` macro provided by `tikzit.sty`. This is essentially a wrapper for `\input`. This macro expects the filename without an extension or `figures/`, so e.g. `\tikzfig{circ}` will input the file `figures/circ.tikz`. You can also use `\ctikzfig` as a shorthand for `\tikzfig` wrapped in the `center` environment. For everything else you ever wanted to know about TikZiT and `tikzit.sty`, check out:

https://tikzit.github.io.

Loading styles in TikZiT

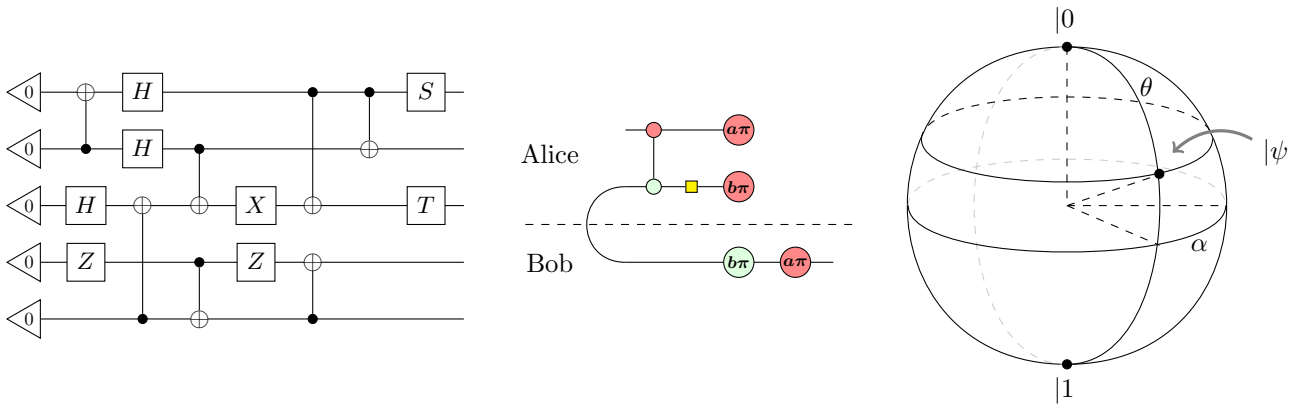
Before you open one of the example figures in TikZiT, make sure you load the `.tikzstyles` file included in this directory. You can do that by clicking the icon that looks like a folder near the top-right of the main TikZiT window. This will give you a handful of styles to start with, and ensure that when you preview figures in TikZiT, they look the same as they do in this paper.

Using pre-built figures

By passing the `[draft]` option to `tikzit.sty`, figures can now be pre-built using the included `Rakefile`, or any other way you like. This can substantially reduce build time if you have lots of figures. With the `draft` option active, `\tikzfig{FOO}` will first search for a file called `cache/FOO.pdf` and include that before trying to build the tikz figure. To use the included build script, you'll need `rake` (`https://github.com/ruby/rake`). Then, you can pre-build simply by running `rake` in from the command line in the same directory as this paper. You can also listen for changes and rebuild figures on-demand with `rake listen`. If you change the name of the main file or add more tex files to your project, edit the appropriate options at the top of the included `Rakefile`.

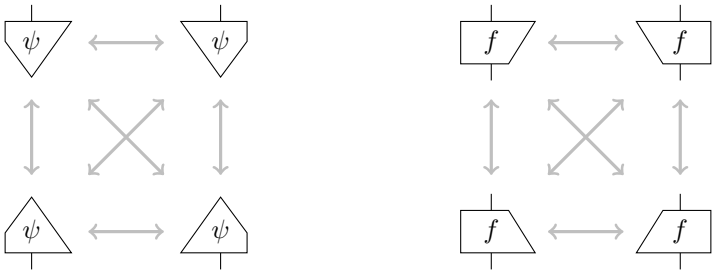
The quantum template

This template contains styles for quantum circuits, ZX-calculus diagrams, and handy stuff like the Bloch sphere:



Asymmetric shapes

This template also provides various asymmetric shapes that are used in the book *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning* to represent the adjoints, transposes, and conjugates of states or maps.



These shapes are defined by hand using PGF code in `quantum.tikzdefs`. For maps, the shapes are called `NWbox`, `NEbox`, `SWbox`, and `SEbox`. The directions NW, NE, SW, and SE indicate which way the asymmetric part of the box "points". Similarly, the shapes for states are called `NWtriangle`, `NEtriangle`, `SWtriangle`, and `SEtriangle`.

To use these shapes, you should create a style and set the `shape` property to be one of the above. There are already example styles using all of these shapes defined in `quantum.tikzstyles`.

Note that TikZiT doesn't recognise custom shapes in the GUI. In practice, this isn't such a problem. The way I get around this is to use certain properties that only affect how a node appears in TikZiT (e.g. `tikzit shape` and `tikzit fill`) to visually distinguish styles using different custom shapes. For example, the styles used in the pictures above are colour-coded in TikZiT. The custom shapes will nevertheless draw properly when using TikZiT's "Preview" feature.