The QWorld Package

Version 1.1.1

Niina Ryota

April 14, 2025

Abstract

`QWorld` is a LaTeX package designed for efficiently rendering complex graphical calculus in monoidal categories. In particular, it supports the typesetting of diagrammatic languages in category theory, quantum theory, and related fields, where diagrammatic reasoning plays a crucial role.

Contents

1 Introduction

This document aims to provide a step-by-step guide to installing and using the `QWorld` package, from basic commands to advanced applications. `QWorld` is a LaTeX package specifically designed for the typesetting of diagrammatic languages, offering an intuitive command set for describing diagrams related to monoidal categories, Frobenius structures, braiding, Hopf algebras, symmetries, dualities, pivotal structures, and dagger categories using Ti*k*Z [1].

2 Basic Usage

2.1 Fundamental Elements

The following are the most fundamental commands in the `QWorld` package, forming the basis for drawing diagrams:

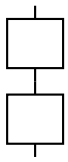
- `\q`: A canvas for placing diagram elements.
- `\qbox`: Draws a box  .
- `\qwire`: Draws a wire .
- `\qcirc`: Compositon.

These commands are used within the canvas environment, denoted by `\q{...}`. For example, writing `\q{\qbox}` results in the rendering of a box  .

To compose  and , use `\qcirc`. This is analogous to function composition in mathematics, such as $g \circ f$ (`\(g\circ f\)`).

Example 2.1:

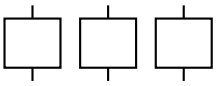
```
\q{ \qbox \qcirc \qbox }
```



Additionally, placing multiple `\qbox` commands in sequence results in boxes  being arranged horizontally:

Example 2.2:

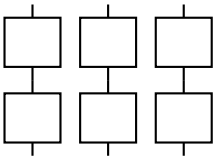
```
\q{ \qbox \qbox \qbox }
```



It is also possible to compose multiple boxes using`\qcirc`:

Example 2.3:

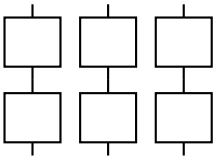
```
\q
{
  \qbox \qbox \qbox
    \qcirc
  \qbox \qbox \qbox
}
```



Note that `\n` is synonymous with`\qcirc`

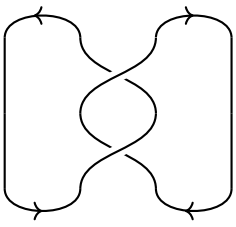
Example 2.4:

```
\q
{
  \qbox \qbox \qbox \n
  \qbox \qbox \qbox
}
```



Example 2.5:

```
\q{
  \qcaprev          \qcap  \n
  \qwire   \qbraid \qwire \n
  \qwire   \qbraid \qwire \n
  \qcup             \qcuprev
}
```



- `\qcaprev` draws .

- `\qcap` draws  .
- `\qwire` draws .
- `\qbraid` draws  .
- `\qcup` draws  .
- `\qcuprev` draws  .

As a whole, the elements are arranged in the diagram according to the order in which the commands are written.

2.2 Mathematical Mode

When writing equations that include diagrammatic language, use the mathematical mode.

Example 2.6:

```
\[
  \operatorname{Tr}\left(\q{\qbraid\qcirc\qbraid}\right)
  =\q{
    \qcap               \qcap    \n
    \qwire   \qbraidinv \qwire   \n
    \qwire   \qbraidinv \qwire   \n
    \qcuprev            \qcuprev
  }
\]
```

$$\operatorname{Tr}\left(\q{\qbraid\qcirc\qbraid}\right)=\q{\qcap\qcap\n\qwire\qbraidinv\qwire\n\qwire\qbraidinv\qwire\n\qcuprev\qcuprev}$$

2.3 Installation

The style file for the `QWorld` package, `qworld.sty`, is available for download from CTAN. To use it, place the file in your TEX package directory and include the following line in the preamble:

```
\usepackage{qworld}
```

Dependencies The package depends on the following:

- Ti*k*Z, with the following libraries:
  - `cd`
  - `positioning`

  - arrows
  - arrows.meta
  - calc
  - intersections
  - shapes.symbols
  - shapes.geometric
  - shapes.misc
  - decorations.pathreplacing
  - decorations.markings
  - decorations.pathmorphing
- pgffor
- ifthen
- xparse
- xfp
- xstring

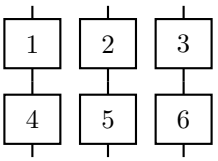
These dependencies are included by default in the TEX Live distribution, so no additional setup is required.

3 Basic Customization

This section explains how to customize commands in the QWorld package.

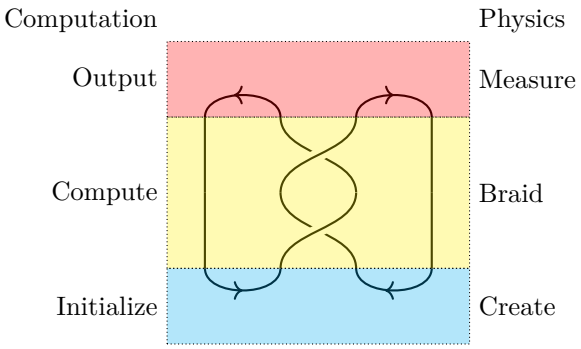
3.1 Identification Number

All commands placed within \q{...} are automatically assigned unique identification numbers based on their execution order. For example, in the previous example 2.4, each \qbox is numbered as follows:

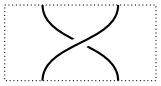


For more details, see Section 3.5 on page 17.

3.2 Bounding Box



Each element has an associated bounding box. The following are examples:

- The bounding box of  is visualized as  .
- The bounding box of  is visualized as  .
- The bounding box of is visualized as  .
- The bounding box of  is visualized as  .

The following commands are available for visualizing bounding boxes and placing text near them:

- Visualization
  - `\qbb`: Displays the bounding box of a specific element.
  - `\qbball`: Displays the bounding boxes of all elements at once.
  - `\qBBall`: Displays the bounding box surrounding the entire canvas.
- Placing text
  - `\bbsymbol`: Places text within a bounding box.

`\qbb[<color>][<ID list>]`

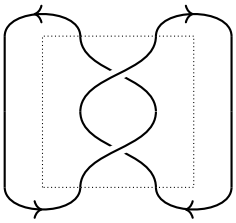
- First argument (optional): Specifies the color. If omitted, no color is applied.
- Second argument: Specifies a list of element IDs to display their bounding boxes.

Example 3.1:

```
\q{\qbox\qbb[1]}
```

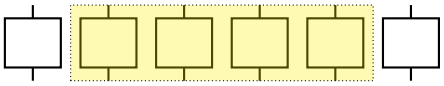
Example 3.2:

```
\q{
  \qcaprev          \qcap  \n
  \qwire   \qbraid \qwire \n
  \qwire   \qbraid \qwire \n
  \qcup             \qcuprev
  \qbb[4,7]
}
```



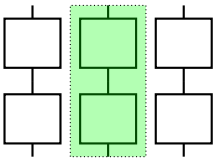
Example 3.3:

```
\q{\qbox\qbox\qbox\qbox\qbox\qbox\qbb[yellow][2,...,5]}
```



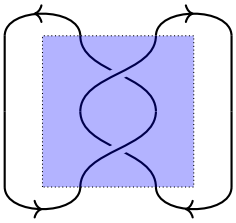
Example 3.4:

```
\q
{
  \qbox\qbox\qbox\n
  \qbox\qbox\qbox
  \qbb[green][2,5]
}
```



Example 3.5:

```
\q{
  \qcaprev \qcap          \n
  \qwire   \qbraid \qwire \n
  \qwire   \qbraid \qwire \n
  \qcup    \qcuprev
  \qbb[blue][4,7]
}
```

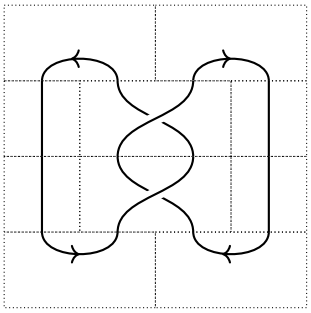


`\qbball[<color>]`

- Displays multiple bounding boxes at once (even for a single element).
- Ensures that the bounding boxes of different elements are properly aligned without overlap.
- The `QWorld` package **minimizes the need for tedious manual coordinate adjustments** in diagram drawing.

Example 3.6:

```
\q{
  \qcaprev          \qcap    \n
  \qwire   \qbraid \qwire    \n
  \qwire   \qbraid \qwire    \n
  \qcup             \qcuprev
  \qbball
}
```

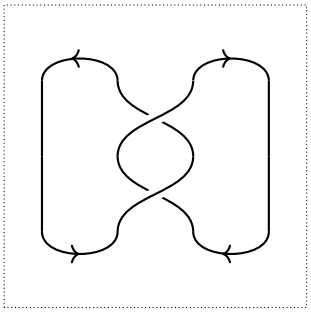


`\qBBall[<color>]`

- Displays the bounding box surrounding all elements within the canvas.

Example 3.7:

```
\q{
  \qcaprev          \qcap    \n
  \qwire   \qbraid \qwire    \n
  \qwire   \qbraid \qwire    \n
  \qcup             \qcuprev
  \qBBall
}
```



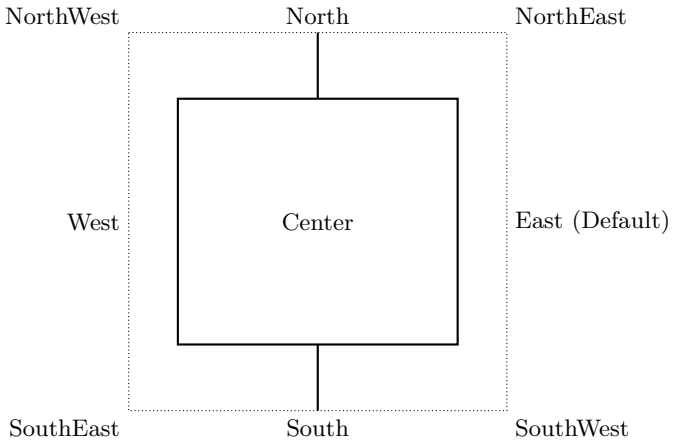
`\bbsymbol[<direction>]{<text>}[<options>]`

- First argument (optional): Specifies placement direction. Options:
  - `N` (North), `S` (South), `E` (East), `W` (West),
  - `NW` (Northwest), `NE` (Northeast), `SE` (Southeast), `SW` (Southwest),
  - `C` (Center).
  - Default: `E` (East).
- Second argument: Specifies the text to be placed.
- Third argument (optional): Adjusts position (e.g., `above`, `below`, `left`, `right`).
  - Default: `right`.

Example 3.8:

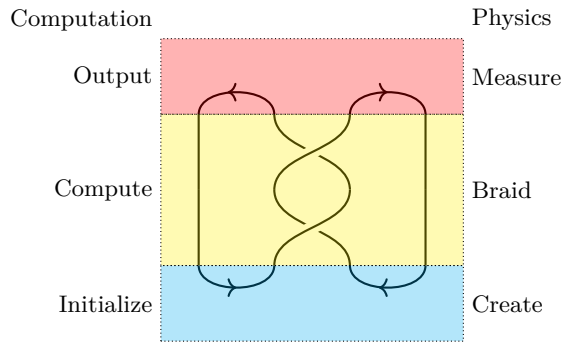
```
\q[scale=5]{
  \qbox
```

```
  \qbb[1]
  \bbsymbol{East (Default)}
  \bbsymbol[W]{West}[left]
  \bbsymbol[N]{North}[above]
  \bbsymbol[S]{South}[below]
  \bbsymbol[NE]{NorthEast}[above right]
  \bbsymbol[NW]{NorthWest}[above left]
  \bbsymbol[SE]{SouthWest}[below right]
  \bbsymbol[SW]{SouthEast}[below left]
  \bbsymbol[C]{Center}[]
}
```



Example 3.9:

```
\q
{
  \qcaprev          \qcap    \n
  \qwire   \qbraid \qwire   \n
  \qwire   \qbraid \qwire   \n
  \qcup             \qcuprev
  \qbb[red][1,2]
  \bbsymbol{Measure}
  \bbsymbol[W]{Output}[left]
  \bbsymbol[NW]{Computation}[above left]
  \bbsymbol[NE]{Physics}[above right]
  \qbb[yellow][3,5,6,8]
  \bbsymbol{Braid}
  \bbsymbol[W]{Compute}[left]
  \qbb[cyan][9,10]
  \bbsymbol{Create}
  \bbsymbol[W]{Initialize}[left]
}
```



3.3 Box Customization

Keys can be used to adjust the number and arrangement of the box's boundary lines. For example, the following boxes can be drawn:

- 
- 

These keys are applicable to many elements other than wires. The main keys are as follows:

- `n`
  - The number of points on the north edge of the box  where wires | can be generated (these points will be referred to as `n` terminals).
  - For example, if `n=3`, three points can be placed at equal intervals on the north edge of the box.
  - The default value is `n=1`.
  - The point obtained by vertically moving an `n` terminal to the northernmost position is called an `N` terminal.
  - Both `n` terminals and `N` terminals are indexed as $1, 2, ...$ from the west (left).
  - Generating a wire from the i-th `n` terminal means connecting the i-th `n` terminal and the i-th `N` terminal with a line segment.
- `N`
  - The set of indices of the `n` terminals from which wires are actually generated.
  - For example, if `n=3` and `N={2}`, a wire can be generated from the 2nd of the three `n` terminals.
  - If `n` is unspecified and `N` is specified, `n` is automatically set to $\texttt{n} := \max \texttt{N}$.
  - If `N` is unspecified and `n` is specified, `N` is automatically set to $\texttt{N} := \{1, ..., \texttt{n}\}$.

- The northernmost end of a wire actually generated from an `n` terminal is called the output point, or `O` terminal.

- `s`
  - The number of points on the south edge of the box where wires can be generated (these points will be referred to as `s` terminals).
  - For example, if `s=4`, four points can be placed at equal intervals on the south edge of the box.
  - The default value is `s=1`.
  - The point obtained by vertically moving an `s` terminal to the southernmost position is called an `S` terminal.
  - Both `s` terminals and `S` terminals are indexed as $1, 2, ...$ from the west (left).
  - Generating a wire from the i-th `s` terminal means connecting the i-th `s` terminal and the i-th `S` terminal with a line segment.

- `S`
  - The set of indices of the `s` terminals from which wires are actually generated.
  - For example, if `s=4` and `S={1,4}`, wires can be generated from the 1st and 4th of the four `s` terminals.
  - If `s` is unspecified and `S` is specified, `s` is automatically set to $\texttt{s} \coloneqq \max \texttt{S}$.
  - If `S` is unspecified and `s` is specified, `S` is automatically set to $\texttt{S} \coloneqq \{1, ..., \texttt{s}\}$.
  - The southernmost end of a wire actually generated from an `s` terminal is called the input point, or `I` terminal.

- `hlen`
  - The horizontal length of the bounding box.
  - For example, if `hlen=2`, the horizontal length of the bounding box becomes `2`.
  - With one exception, the distance between the westernmost wire and the bounding box's boundary, and the distance between the easternmost wire and the bounding box's boundary, are both 0.5.
  - When `hlen` is unspecified, the `n` terminals and `s` terminals are arranged with an interval of 1.
  - When `hlen` is specified, the `n` terminals and `s` terminals are arranged at equal intervals in a section of length $\texttt{hlen} - 1$. The spacing depends on `hlen` and the number of terminals.
    - If there is only one terminal, it is placed at the center of the section. In this case, if $\texttt{hlen} > 1$, the distance between the westernmost wire and the bounding box's boundary, and the distance between the easternmost wire and the bounding box's boundary, are both $\frac{\texttt{hlen}}{2} > 0.5$.

- `vlen`
  - The vertical length (height) of the bounding box.
  - The default value is `vlen=1`.
- `id`
  - The identifier of the box.
  - For example, if `id=ID`, the following names are automatically assigned to the various points:
    * The i-th `n` terminal (counting from the west): (n-i-ID).
    * The i-th `s` terminal: (s-i-ID).
    * The point obtained by vertically moving the i-th `n` terminal to the north edge of the box: (N-i-ID).
    * The point obtained by vertically moving the i-th `s` terminal to the south edge of the box: (S-i-ID).
    * The northwest corner of the bounding box: (NW-ID)
    * The northeast corner of the bounding box: (NE-ID)
    * The southwest corner of the bounding box: (SW-ID)
    * The southeast corner of the bounding box: (SE-ID)
    * The northernmost point of the bounding box: (N-ID)
    * The southernmost point of the bounding box: (S-ID)
    * The westernmost point of the bounding box: (W-ID)
    * The easternmost point of the bounding box: (E-ID)
    * The center of the bounding box: (C-ID)
    * The northwest corner of the box: (nw-ID)
    * The northeast corner of the box: (ne-ID)
    * The southwest corner of the box: (sw-ID)
    * The southeast corner of the box: (se-ID)
    * The northernmost point of the box: (n-ID)
    * The southernmost point of the box: (s-ID)
    * The westernmost point of the box: (w-ID)
    * The easternmost point of the box: (e-ID)
    * The i-th output point: (O-i-ID)

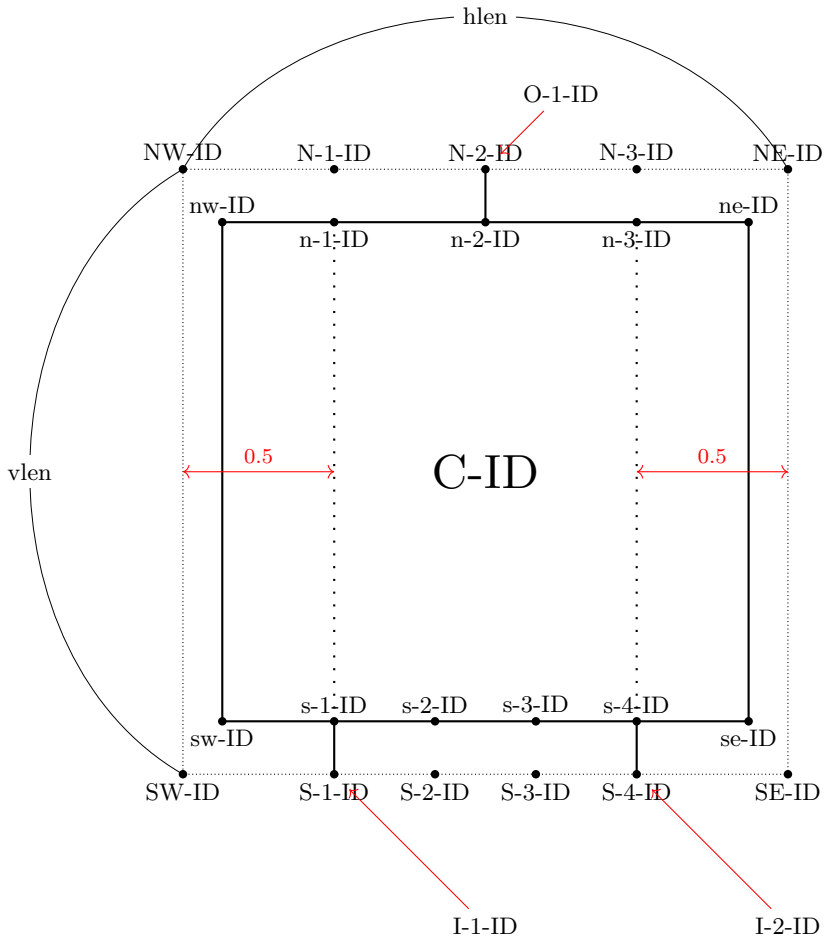
    - The i-th input point: (I-i-ID)
  - Regardless of whether it is specified, each command placed inside `\q{...}` is assigned an identification number. Replacing the ID part with that number refers to the same point. See section 3.5 for details.

- `p`, `P`
  - For a positive integer `k`, `p=k` means `n=k` and `s=k`.
  - For a finite set of positive integers `K`, `P=K` means `N=K` and `S=K`.
- `name`
  - With `name=<text>`, <text> is placed at the center of the box.

Example 3.10:

```
\def\pradius{0.4pt}
\q[scale=4]
{
  \qbox
  [
    n=3,N={2},
    s=4,S={1,4},
    hlen=2,
    vlen=2,
    id={ID},
    name={C-ID}
  ]\n
  \qbb[ID]
  \node [above] at (NE-ID) {NE-ID};
  \node [above] at (NW-ID) {NW-ID};
  \node [below] at (SE-ID) {SE-ID};
  \node [below] at (SW-ID) {SW-ID};
  \fill (NE-ID) circle (\pradius);
  \fill (NW-ID) circle (\pradius);
  \fill (SE-ID) circle (\pradius);
  \fill (SW-ID) circle (\pradius);
  \node [above] at (ne-ID) {ne-ID};
  \node [above] at (nw-ID) {nw-ID};
  \node [below] at (se-ID) {se-ID};
  \node [below] at (sw-ID) {sw-ID};
  \fill (ne-ID) circle (\pradius);
  \fill (nw-ID) circle (\pradius);
  \fill (se-ID) circle (\pradius);
  \fill (sw-ID) circle (\pradius);
  \foreach\i in {1,...,3}
  {
    \symboln[\i]{n-\i-ID}[below]
    \fill (n-\i-ID) circle (\pradius);
    \symbolN[\i]{N-\i-ID}
    \fill (N-\i-ID) circle (\pradius);
  }
```

```
  \foreach\i in {1,...,4}
  {
    \symbols[\i]{s-\i-ID}[above]
    \fill (s-\i-ID) circle (\pradius);
    \symbolS[\i]{S-\i-ID}[below]
    \fill (S-\i-ID) circle (\pradius);
  }
  \draw[line width=0.2pt] (NW-ID)
    to [bend left=60] node [fill=white, midway] {hlen} (NE-ID);
  \draw[line width=0.2pt] (NW-ID)
    to [bend right=60] node [fill=white, midway] {vlen} (SW-ID);
  \draw [red, <->] (C-ID -| W-ID)
    -- node [midway, auto, font=\footnotesize] {0.5} (C-ID -| S-1-ID);
  \draw [red, <->] (C-ID -| S-4-ID)
    -- node [midway, auto, font=\footnotesize] {0.5} (C-ID -| E-ID);
  \qwire[arrowtype={loosely dotted}, dom={s-1-ID}, cod={n-1-ID}]
  \qwire[arrowtype={loosely dotted}, dom={s-4-ID}, cod={n-3-ID}]
  \node (O1prime) at ([shift={(0.25,0.25)}]O-1-ID){O-1-ID};
  \draw[->,red] (O1prime) to ([shift={(0.05,0.05)}]O-1-ID);
  \node (I1prime) at ([shift={(0.5,-0.5)}]I-1-ID){I-1-ID};
  \node (I2prime) at ([shift={(0.5,-0.5)}]I-2-ID) {I-2-ID};
  \draw[->,red](I1prime) to ([shift={(0.05,-0.05)}]I-1-ID);
  \draw[->,red](I2prime) to ([shift={(0.05,-0.05)}]I-2-ID);
}
```



The `\qwire arrowtype` key, as well as the `dom` and `cod` keys, are explained in section 3.9 on page 23.

Label (`\symbol`)

```
\symbolI[<number>]{<label>}[<options>]
```

For example, `\symbolI[2]{text}[below right, id=ID]` places the label "text" at the bottom right of the second input point (I-2-ID) of `\q{\qbox[p=3, id=ID]}` ( ). When no number or options are specified, the default values 1, below, and `id=<previous>` are applied, respectively. Other label display commands include:

```
\symbolS[<number>]{<label>}[<options>]
```

```
\symbols[<number>]{<label>}[<options>]
```

```
\symbolO[<number>]{<label>}[<options>]
```

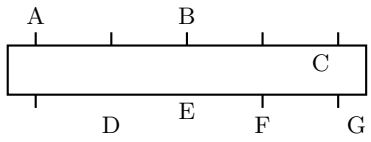
`\symbolN[<number>]{<label>}[<options>]`

`\symboln[<number>]{<label>}[<options>]`

`\symbolS` and `\symbols` place the label <label>at points (S-i-ID) and (s-i-ID), respectively. The default number i is 1, and the default option is below. `\symbolO`, `\symbolN`, and `\symboln` place the label <label>at points (O-i-ID), (N-i-ID), and (n-i-ID), respectively. The default number i is 1, and the default option is above.

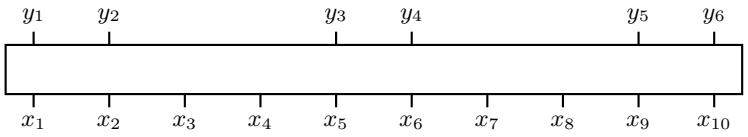
Example 3.11:

```
\q{
  \qbox[p=5, S={1,4,5}, id=ID]
  \symbolO{A}
  \symbolN[3]{B}
  \symboln[5]{C}[below left]
  \symbolS[2]{D}
  \symbols[3]{E}
  \symbolI[2]{F}
  \symbolI[3]{G}[below right]
}
```



Example 3.12:

```
\q{
  \qbox[p=10,N={1,2,5,6,9,10}, id=ID]
  \foreach\i in {1,...,10}
  {
    \symbolI[\i]{\(x_{\i}\)}
  }
  \foreach\j in {1,...,6}
  {
    \symbolO[\j]{\(y_{\j}\)}
  }
}
```



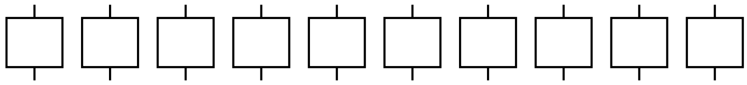
3.4 Repetition

`\qloop[<number of repetitions>]{<command>}`

By using the `\qloop` command, you can repeat a command the specified number of times:

Example 3.13:

```
\q{ \qloop[10]{\qbox} }
```



The \qloop command can also be used outside of \q:

Example 3.14:

```
\qloop[30]{I AM BOB}
```

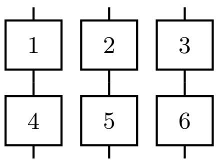
I AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB I
AM BOB I AM BOB I AM BOB

3.5 Displaying Identification Numbers

Commands inside \q{...} are automatically assigned identification numbers. If `show id=true` is specified, those identification numbers will be displayed.

Example 3.15:

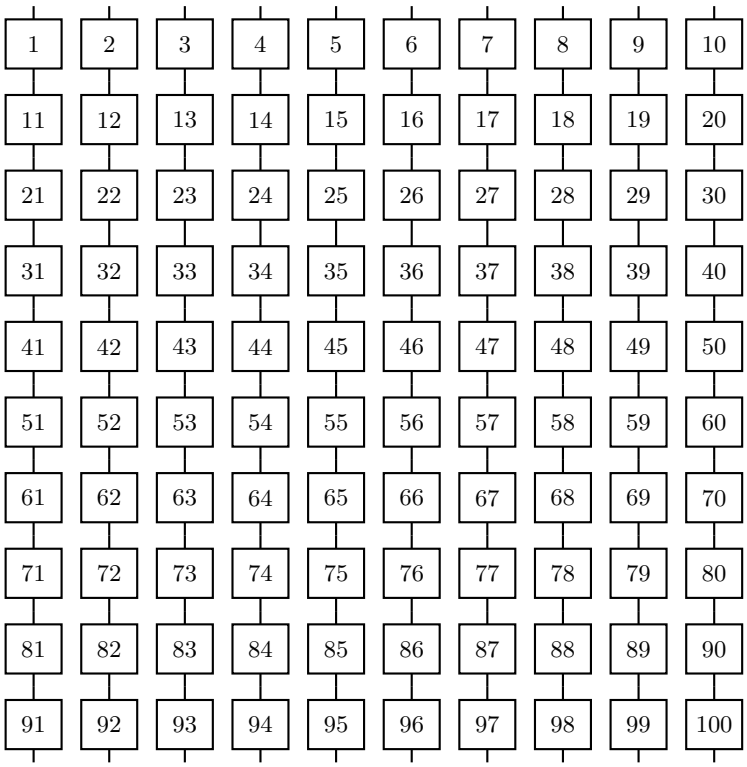
```
\q{
  \qbox[show id={true}] \qbox[show id={true}] \qbox[show id={true}]\n
  \qbox[show id={true}] \qbox[show id={true}] \qbox[show id={true}]
}
```



Example 3.16:

```
\q{
  \qloop[10]
  {
    \qloop[10]
    {
      \qbox[show id={true}]
```

```
    }\n
  }\n
}
```



3.6 Color

You can specify colors for the boxes:

- `color`
  - The color inside the box, for example, `color=green` will produce  .
- `frame color`
  - The color of the box's frame, for example, `frame color=cyan` will produce  .
- `morphism color`
  - The color of the morphisms inside the box, for example, `morphism color=red` will produce  .

Example 3.17:

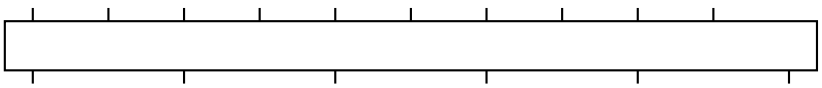
```
\q[scale={\fpeval{pi}}]{
  \qbox
  [
    name={\textbf{Q-WORLD}},
    color=red,
    frame color={green!50!black},
    morphism color=yellow
  ]
}
```

N, S, Arithmetic Sequence `QWorld` loads the `pgffor` package, so the shorthand notation using `...` can be used for lists:

- $1, ..., 10$: An arithmetic sequence with a common difference of 1
- $1, 3, ..., 11$: An arithmetic sequence based on the first two terms with a common difference

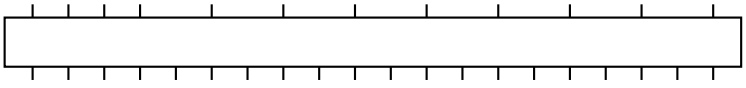
Example 3.18:

```
\q{
  \qbox[N={1,...,10}, S={1,3,...,11}]
  }
```



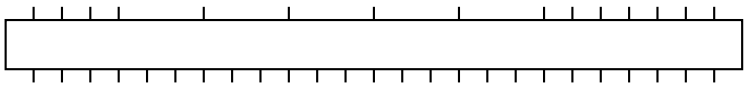
Example 3.19:

```
\q{
  \qbox[N={1,...,4,6,8,...,20}, S={1,...,20}, hlen=10]
}
```



Example 3.20:

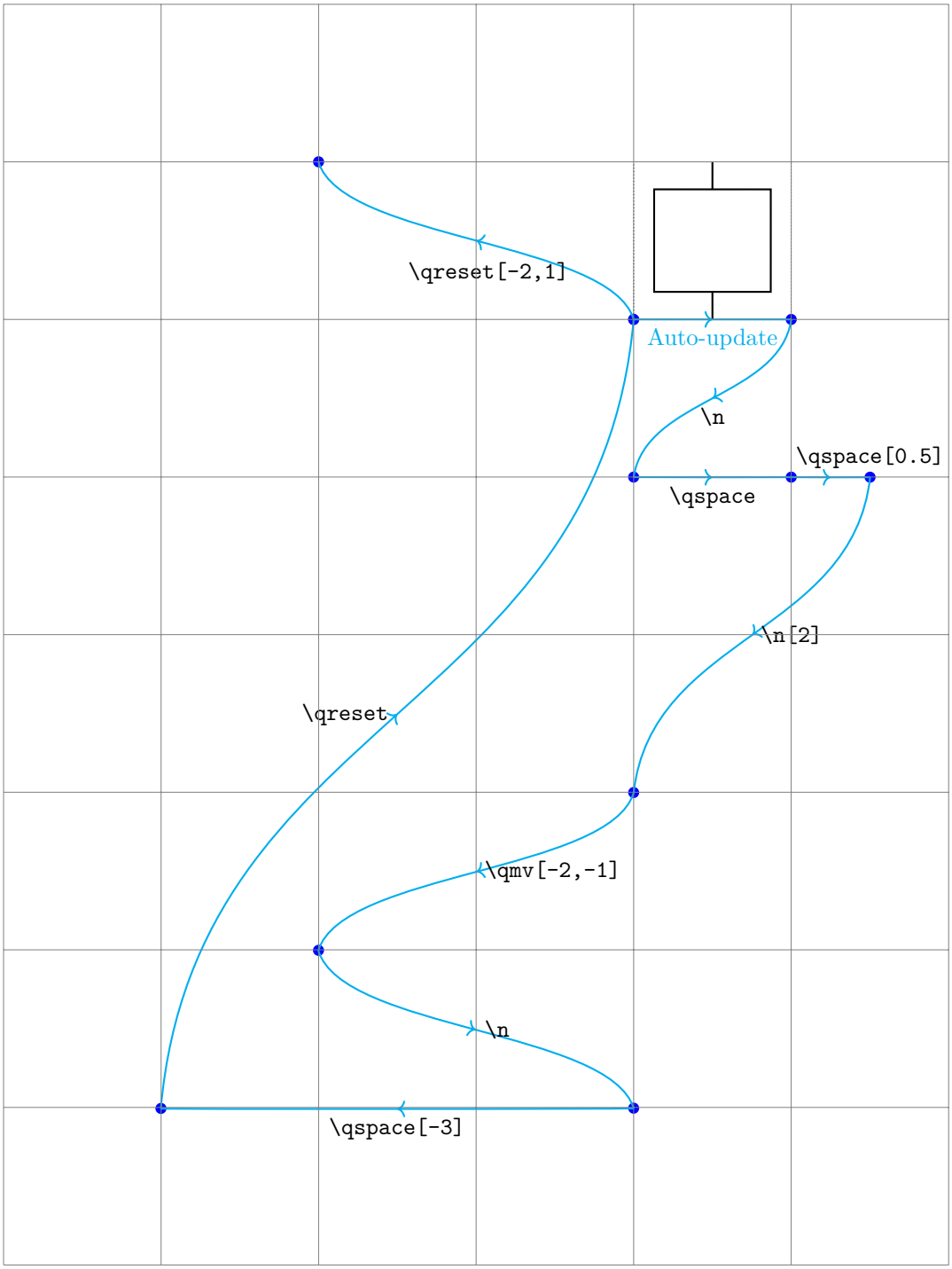
```
\q{
  \qbox[N={1,...,4,7,10,...,19,20,21,...,25}, S={1,...,25}, hlen=10]
}
```



3.7 Relative Coordinates

This is a command for specifying the position of elements relative to a given point. Note that x and y do not have to be integers; expressions evaluable by `\fpeval`, such as `2^1.5`, `cos(pi)`, and `sprt(2)`, are valid.

- `\qmv[`x`,`y`]`
  - Translates by (x, y).
  - The default is $x = y = 0$.
- `\n[`y`]`
  - Moves vertically by y ($Y \mapsto Y - y$), and sets the X coordinate to 0.
  - The default is $y = 1$, which behaves like a line break.
- `\qspace[`x`]`
  - Moves Horizontally by x.
  - The default is $x = 1$, which behaves like the spacebar.
  - This is equivalent to `\qmv[`x`,0]`.
- `\qreset[`x`,`y`]`
  - Moves to the point (x,y).

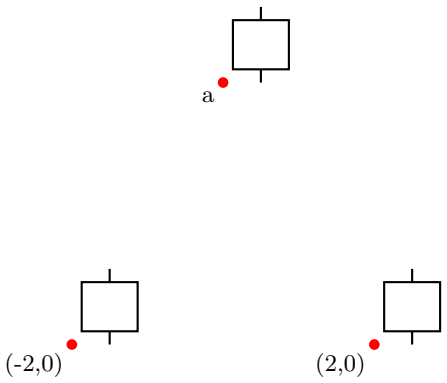


3.8 Absolute Coordinates

By using the `at` key, positions can be specified with absolute coordinates. `\q{\qbox[at={2,3}]}` is equivalent to `\q{\qreset[2,3]\qbox}`.

Example 3.21: Absolute Coordinates

```
\q{
  \coordinate (a) at (0,{\fpeval{2*sqrt(3)}});
  \qbox[at=a]
  \qbox[at={-2,0}]
  \qbox[at={2,0}]
  \fill[red] (a) circle (2pt);
  \fill[red] (-2,0) circle (2pt);
  \fill[red] (2,0) circle (2pt);
  \node[below left] at (a) {a};
  \node[below left] at (-2,0) {(-2,0)};
  \node[below left] at (2,0) {(2,0)};
}
```



The TEX language is Turing complete[1]. In other words, every algorithm that exists in the world can be expressed in TEX. Phisical simulations (such as weather forecasting or chemical reaction prediction), economic simulations (such as investment strategies), and even artificial intelligence learning algorithms (such as image recognition or natural language processing)——

all of these can be implemented in TEX.

However, how many people would actually want to do that? I for one, wouldn't want to write even a sorting algorithm in TEX. That's like tring to create Super Mario in spreadsheet.

However, there is a twist.

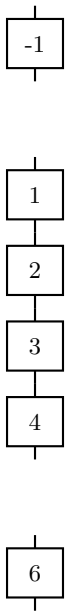
I had a "revelation"[2]. And with the `at` key, I created a sorting algorithm.

[1] For example, the ability to simulate a Turing machine is demonstrated in the example from LiteratePrograms[4]. Additionally, an article on Overleaf introduces a method to compute Fibonacci numbers using TEX, illustrating the practical utility of its Turing completeness [5]

[2] Original source: On January 20, 2011, an algorithm called "Sleep Sort" was posted on the anonymous message board 4chan [2, 3]. This algorithm uses a `Bash` script to execute each input value as a thread asynchronously, sorting the list by the timing of each thread's completion (the specified "sleep" time).

Example 3.22:

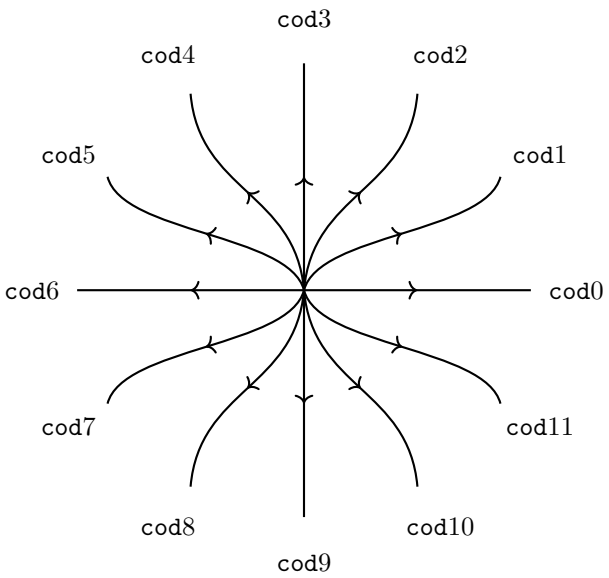
```
\newcommand{\AtSort}[1][1]{
  \q{
    \foreach \i in {#1}
    {
      \qbox[name={\i}, at={0, -\i}]\n
    }
  }
}
\AtSort[4,1,3,-1,6,2]
```



It works. Indeed, it works.

But there is one problem. **If the list contains elements with large absolute values, the output will overflow the page**.

3.9 Wire Customization

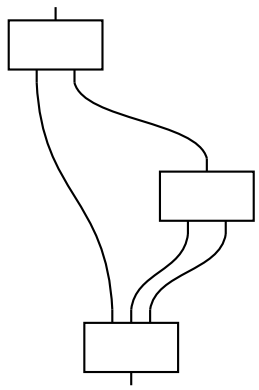


Among the keys of the box  introduced in section 3.3, `id` and `vlen` are also valid for wires. The basic keys specific to wires are as follows:

- `dom`: input point.
- `cod`: output point.
  - Default value is the position obtained by moving `dom` 1 unit vertically to the north.
- `label`: Label for the wire.
- `label at`: Position of the label on the wire.
  - The default is midway.
- `label side`: Relationship between the label and the wire.
  - The default is right.

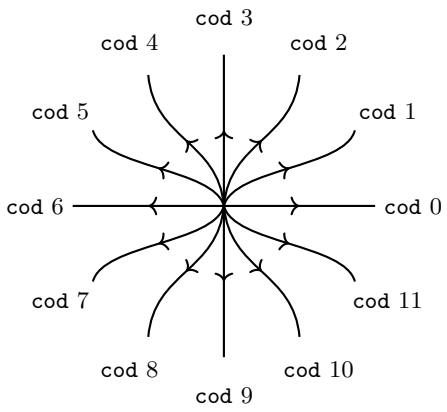
Example 3.23:

```
\q{
  \qbox[n=3,hlen=1.5]
  \qbox[at={1,2}, s=2, hlen=1.5]
  \qbox[at={-1,4}, s=2, hlen=1.5]
  \qwire[dom={N-1-1}, cod={S-1-3}]
  \qwire[dom={N-2-1}, cod={S-1-2}]
  \qwire[dom={N-3-1}, cod={S-2-2}]
  \qwire[dom={N-1-2}, cod={S-2-3}]
}
```



Example 3.24:

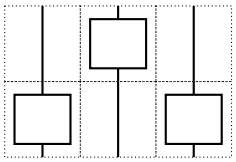
```
\q{
  \def\maxi{11}
  \foreach \i in {0,...,\maxi}
  {
      \qwireup[dom={0,0}, cod={{\fpeval{\i*360/(\maxi+1)}}:2}]
      \node at ({\fpeval{\i*360/(\maxi+1)}}:2.5) {\texttt{cod} \i};
  }
}
```



The bounding box of the wire is the same as that of  :

Example 3.25:

```
\q{
  \qwire\qbox\qwire\n
  \qbox\qwire\qbox
  \qbball
}
```

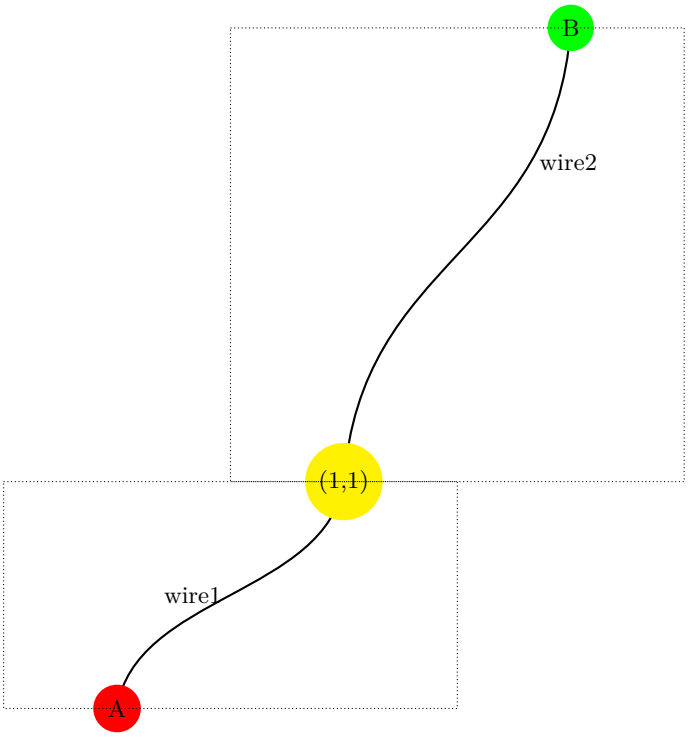


The bounding box of the wire matches that of a box with `dom` as the input point and `cod` as the output point:

- and 
-  and 

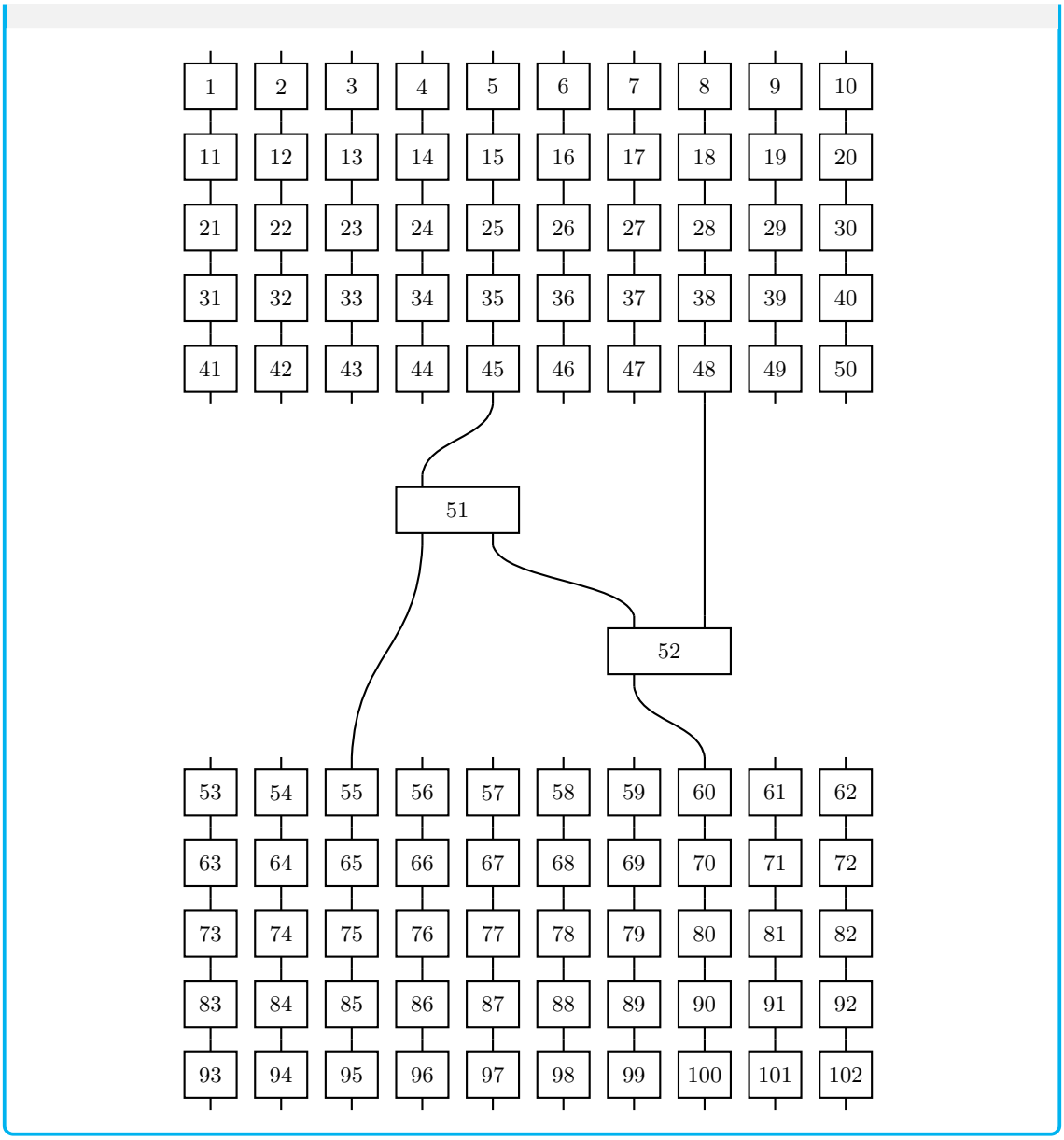
Example 3.26:

```
\q[scale=3]{
  \coordinate (A) at (0,0);
  \coordinate (B) at (2,3);
  \qwire[dom=A, cod={1,1}, label={wire1}, label side={left}]
  \qwire[dom={1,1}, cod=B, label={wire2}, label at={near end}]
  \node[fill=red, circle] at (A) {A};
  \node[fill=green, circle] at (B) {B};
  \node[fill=yellow, circle] at (1,1) {(1,1)};
  \qbball
}
```



Example 3.27:

```
\q{
  \qloop[5]{
    \qloop[10]{
      \qbox[show id={true}]
    }\n
  }\n
  \qspace[3]\qbox[s=2,show id={true},id={ego1}]\n\n
  \qspace[6]\qbox[n=2,show id={true},id={ego2}]\n\n
  \qloop[5]{
    \qloop[10]{
      \qbox[show id={true}]
    }\n
  }
  \qwire[dom={O-1-55},cod={I-1-51}]
  \qwire[dom={O-1-60},cod={I-1-52}]
  \qwire[dom={O-1-51},cod={I-1-45}]
  \qwire[dom={O-1-52},cod={I-2-51}]
  \qwire[dom={O-2-52},cod={I-1-48}]
}
```



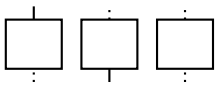
arrowtype The wire style is specified using the `arrowtype` key.

Example 3.28:

```
\q{
  \qwire[arrowtype=dotted]
  \qwire[arrowtype=dashed]
  \qwire[arrowtype=dash dot]
  \qwire[arrowtype=dash dot dot]
  \qwire[arrowtype=densely dotted]
  \qwire[arrowtype=loosely dotted]
}
```

Example 3.29:

```
\q{
  \qbox[dom arrowtype=dotted]
  \qbox[cod arrowtype=dotted]
  \qbox[arrowtype=dotted]
}
```



Example 3.30:

```
\q{
  \qspider[dom arrowtype=dotted]
  \qspider[cod arrowtype=dotted]
  \qspider[arrowtype=dotted]
}
```

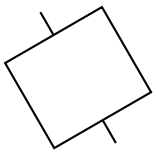
3.10 Canvas

```
\q[option]
{
  command 1
  command 2
  ...
}
```

The entire diagram can be transformed useing canvas options.

Example 3.31:

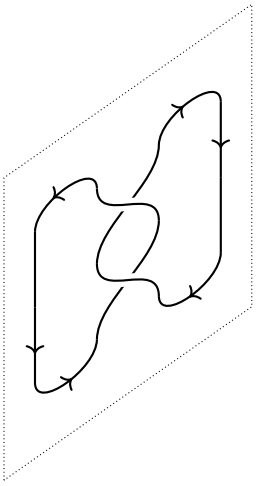
```
\q[scale=2,rotate=30]{\qbox}
```



It is also possible to **slant** the canvas.

Example 3.32:

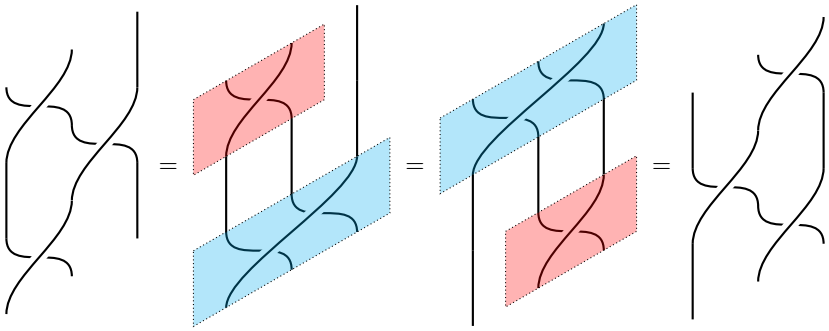
```
\def\anglerot{-55}
\tikzset
{
  my slant style/.style={
    yslant=-cot(\anglerot),
    xscale=sin(-\anglerot)
    }
}
\q[my slant style]
{
  \qcaprev\qcap\n
  \qwire\qbraidinv\qwiredown\n
  \qwiredown\qbraidinv\qwire\n
  \qcup\qcuprev
  \qBBall
}
```



Example 3.33:

```
\def\anglerot{-60}
\tikzset
{
  my slant style/.style={
    yslant=-cot(\anglerot),
    xscale=sin(-\anglerot)
  }
}
\[
  \q[my slant style]{
    \qbraid\qwire\n
    \qwire\qbraid\n
```

```
    \qbraid\qwire
  }=\q[my slant style]{
    \qbraid\qwire\n
    \qwire\qwire\qwire\n
    \qbraid[num R=2]
    \qbb[red][1]
    \qbb[cyan][6]
  }=\q[my slant style]{
    \qbraid[ num R=2]\n
    \qwire\qwire\qwire\n
    \qwire\qbraid
    \qbb[red][6]
    \qbb[cyan][1]
  }=\q[my slant style]{
    \qwire\qbraid\n
    \qbraid\qwire\n
    \qwire\qbraid
  }
\]
```



4 Monoidal Categories and `QWorld`

In this section, we explore how to typeset graphical calculus using the `QWorld` package, examining its different applications and practical usage.

While `QWorld` can handle diagrams independently, it is often beneficial to combine it with packages like `amsmath`, `amssymb`, and `mathtools` for more flexible mathematical expressions. This section demonstrates how to typeset actual graphical calculus in TEX while introducing the functionalities of `QWorld`.

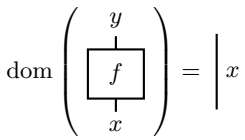
To reproduce the examples in this section, please ensure that the following packages are loaded:

```
\usepackage{amsmath, amssymb, mathtools}
\usepackage{qworld}
```

4.1 Category

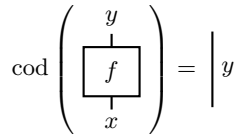
Example 4.1: Domain

```
\[
  \operatorname{dom}\left(
    \q{
      \qbox[name={\(f\)}]
      \symbolI{\(x\)}
      \symbolO{\(y\)}
    }
  \right)=\q{\qwire[label={\(x\)}]}
\]
```



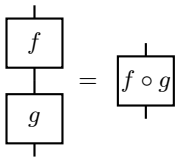
Example 4.2: Codomain

```
\[
  \operatorname{cod}\left(
    \q{
      \qbox[name={\(f\)}]
      \symbolI{\(x\)}
      \symbolO{\(y\)}
    }
  \right)=\q{\qwire[label={\(y\)}]}
\]
```



Example 4.3: (Vertical) Compositon

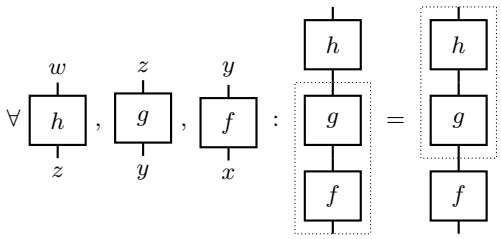
```
\[
  \q{
    \qbox[name={\(f\)}]\qcirc\qbox[name={\(g\)}]
  }=\q{\qbox[name={\(f\circ g\)}]}
\]
```



Example 4.4: Associativity

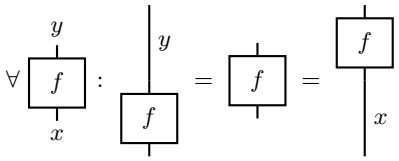
```
\[
  \forall\q{\qbox[name={\(h\)}] \symbolI{\(z\)} \symbolO{\(w\)}},
         \q{\qbox[name={\(g\)}] \symbolI{\(y\)} \symbolO{\(z\)}},
         \q{\qbox[name={\(f\)}] \symbolI{\(x\)} \symbolO{\(y\)}}:
  \q{
      \qbox[name={\(h\)}]\qcirc\qbox[name={\(g\)}]\qcirc\qbox[name={\(f\)}]
      \qbb[2,3]
  }
  =
  \q{
      \qbox[name={\(h\)}]\qcirc\qbox[name={\(g\)}]\qcirc\qbox[name={\(f\)}]
      \qbb[1,2]
  }
```

```
\]
```



Example 4.5: Identity Law

```
\begin{equation}
  \label{law:identity}
  \forall\q
  {
    \qbox[name={\(f\)}]
    \symbolS{\(x\)}
    \symbolN{\(y\)}
  }\colon
  \q{
    \qwire[label={\(y\)}]\n
    \qbox[name={\(f\)}]
  }=\q{
    \qbox[name={\(f\)}]
  }=\q{
    \qbox[name={\(f\)}]\n
    \qwire[label={\(x\)}]
  }
\end{equation}
```



(1)

4.1.1 Object Information and Color

In categorical string diagrammatics, identity morphism are typically represented as simple wires. To make it visually evident which object an identity morphism corresponds to, a method has been introduced that assigns a **distinct color to each object**. This allows object information to be encoded directly on the wire, providing intuitive visual cues for distinguishing between objects-especially in diagrams composed entirely of curved structures. The theoretical foundation of this approach lies in the categorical fact that **every morphism has a uniquely defined domain and codomain object**. Given that this structure is an intrinsic part of any category, it is entirely justified to represent objects diagrammatically via the visual attribute of "color". In diagrams

involving braidings (see Section 4.5.4, page 53) or duality structures (see Section 4.7.3, page 72), color serves as an effective aid in supporting accurate interpretation.

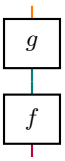
From this perspective, an object may be regarded as something that manifests as a color decorating a wire. The following example illustrates how the domain and codomain colors of morphisms contribute to visually validating the composability of morphisms:

Example 4.6:

```
\(\operatorname{dom}\) is a function that identifies the domain color
\(\q{\qwire[color=red]}\)
of the morphism
\(\q{\qbox[name={\(f\)}, dom color=red, cod color=cyan]}\),
and \(\operatorname{cod}\) similarly identifies the codomain color
\(\q{\qwire[color=cyan]}\).\par
On the page, the codomain color of
\(\q{\qbox[name={\(f\)}, dom color=purple, cod color=teal]}\)
and the domain color of
\(\q{\qbox[name={\(g\)}, dom color=teal, cod color=orange]}\)
are explicitly shown to be the same via color matching.
Only when this condition holds is the vertical composition
\[
  \q{
    \qbox[name={\(g\)}, dom color=teal, cod color=orange]
    \n
    \qbox[name={\(f\)}, dom color=purple, cod color=teal]
  }
\]
accepted as a type-consistent operation.
```

dom is a function that identifies the domain color  of the morphism  , and cod similarly identifies the codomain color  .

On the page, the codomain color of  and the domain color of  are explicitly shown to be the same via color matching. Only when this condition holds is the vertical composition

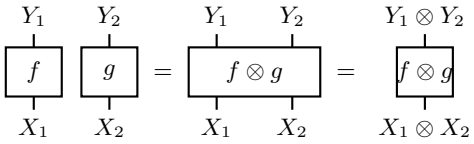


accepted as a type-consistent operation.

4.2 Monoidal Category

Example 4.7: Tensor Product, Horizontal (Parallel) Compositon

```
\[
  \q
  {
    \qbox[name={\(f\)}] \qbox[name={\(g\)}]
    \symbolO{\(Y_1\)}[id=1] \symbolO{\(Y_2\)}
    \symbolI{\(X_1\)}[id=1] \symbolI{\(X_2\)}
  }
  =\q
  {
    \qbox[name={\(f\otimes g\)}, p=2]
    \symbolO{\(Y_1\)} \symbolO[2]{\(Y_2\)}
    \symbolI{\(X_1\)} \symbolI[2]{\(X_2\)}
  }
  =\q
  {
    \qbox[name={\(f\otimes g\)}]
    \symbolO{\(Y_1\otimes Y_2\)}
    \symbolI{\(X_1\otimes X_2\)}
  }
\]
```

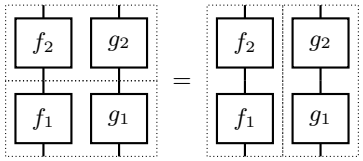


4.2.1 Interchange Law

Example 4.8: Interchange Law (Covariant Functoriality of $\otimes$)

```
\begin{equation}
  \label{law:interchange}
  \q
  {
    \qbox[name={\(f_2\)}]\qbox[name={\(g_2\)}]\n
    \qbox[name={\(f_1\)}]\qbox[name={\(g_1\)}]
    \qbb[1,2]
    \qbb[3,4]
  }
  =\q
  {
    \qbox[name={\(f_2\)}]\qbox[name={\(g_2\)}]\n
    \qbox[name={\(f_1\)}]\qbox[name={\(g_1\)}]
    \qbb[2,4]
    \qbb[1,3]
```

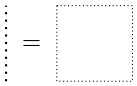
```
  }
\end{equation}
```



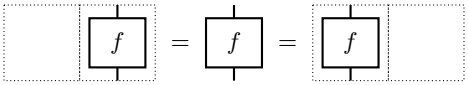
(2)

Example 4.9: Unit Object

```
\begin{gather}
  \q{ \qwire[arrowtype=dotted]}
  =\q{ \qunitob \qbball }\\
  \q{ \qunitob \qbox[name={\(f\)}] \qbball }
  =\q{ \qbox[name={\(f\)}] }
  =\q{ \qbox[name={\(f\)}] \qunitob \qbball }\label{cond:unitob}
\end{gather}
```



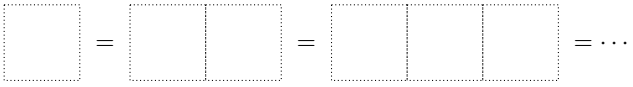
(3)



(4)

Example 4.10:

```
\[
  \q{\qunitob\qbball}
  =\q{\qloop[2]{\qunitob}\qbball}
  =\q{\qloop[3]{\qunitob}\qbball}
  =\cdots
\]
```



4.3 Special Boxes

- `\qstate`:  (Zero Input).
- `\qeffect`:  (Zero Output).
- `\qscalar`:  (Zero Input, Zero Output).

- `\qasym`:  (Asymmetric Box).
- `\qtrans`:  (Transpose Box). Section 4.6.1 on page 55.
- `\qadj`:  (Adjoint Box). Section 4.8.1 on page 73.
- `\qconj`:  (Conjugate Box). Section 4.8.2 on page 74.

4.3.1 Zero Input

Example 4.11:

```
\q{\qstate}
```

4.3.2 Zero Output

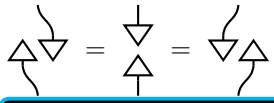
Example 4.12:

```
\q{\qeffect}
```

4.3.3 Zero Input, Zero Output

Example 4.13:

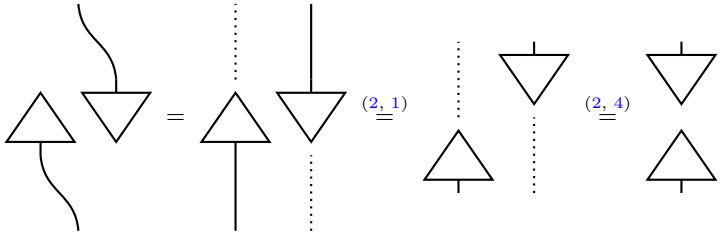
```
\q{\qscalar}
```



Example 4.14:

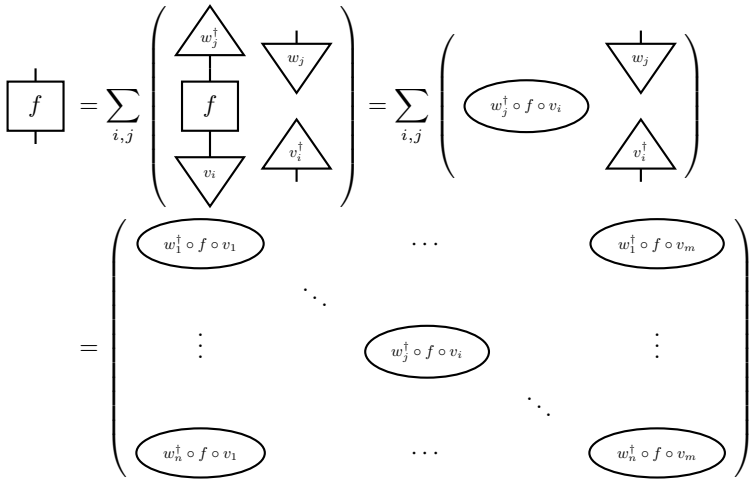
```
\[
  \q{
    \qeffect\qstate
    \qwire[dom={0-1-2}, cod={1,2}]
    \qwire[dom={1,-1}, cod={I-1-1}]
  }=\q{
    \qwire[arrowtype=dotted]\qwire\n
    \qeffect\qstate\n
    \qwire\qwire[arrowtype=dotted]
  }\stackrel{\text{(\ref{law:interchange}, \ref{law:identity})}}{=}\q{
    \qwire[arrowtype=dotted]\qstate\n
    \qeffect\qwire[arrowtype=dotted]
  }\stackrel{\text{(\ref{law:interchange},
  \ref{cond:unitob})}}{=}\q{\qstate\n\qeffect}
```

```
\]
```



Example 4.15: Matrix Representation

```
\begin{align*}
  \q{\qbox[name={\(f\)}]}
  &=\sum\limits_{i,j}
  \left(
    \q{
      \qeffect[name={\(w_j^{\dagger}\)}]\n
      \qbox[name={\(f\)}]\n
      \qstate[name={\(v_i\)}]\n
    }\q{
      \qstate[name={\(w_j\)}]\n
      \qeffect[name={\(v_i^{\dagger}\)}]
    }
  \right)
  =\sum\limits_{i,j}\left(
    \q{
      \qscalar[name={\(w_j^{\dagger}\circ f\circ v_i\)}]
    }
    \q{
      \qstate[name={\(w_j\)}]\n
      \qeffect[name={\(v_i^{\dagger}\)}]
    }
  \right)\\
  &=\begin{pmatrix}
    \q{\qscalar[name={\(w_1^{\dagger}\circ f\circ
  v_1\)}]}&&\cdots&&\q{\qscalar[name={\(w_1^{\dagger}\circ f\circ v_{m}\)}]}\\
    &\ddots&&&\\
    \vdots& &\q{\qscalar[name={\(w_j^{\dagger}\circ f\circ v_i\)}]}& &\vdots\\
    &&&\ddots&\\
    \q{\qscalar[name={\(w_n^{\dagger}\circ f\circ
  v_1\)}]}&&\cdots&&\q{\qscalar[name={\(w_n^{\dagger}\circ f\circ v_m\)}]}
  \end{pmatrix}
\end{align*}
```

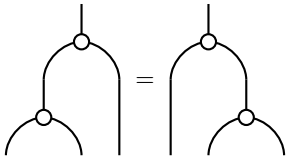


4.4 Monoid Object

- \qmul:
  - hlen=2:
- \qcomul:
  - hlen=2:
- \qunit:
- \qcounit:
- \qspider:

Example 4.16: Associativity

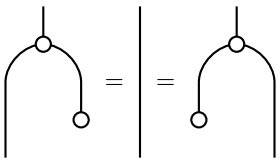
```
\begin{equation}
  \label{law:associate}
  \q{
    \qspace[0.5]\qmul[hlen=2]\n
    \qmul[hlen=2]\qspace[-0.5]\qwire
  }=\q{
    \qmul[hlen=2]\n
    \qwire\qspace[-0.5]\qmul[hlen=2]
  }
\end{equation}
```



(5)

Example 4.17: Unitality

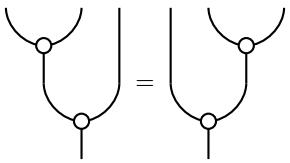
```
\begin{equation}
\label{law:unit}
\q{
  \qmul[hlen=2]\n
  \qwire\qunit
}=\q{\qwire[vlen=2]}
=\q{
  \qmul[hlen=2]\n
  \qunit\qwire
}
\end{equation}
```



(6)

Example 4.18: Coassociativity

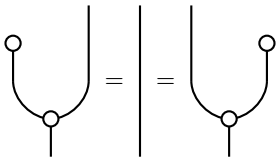
```
\begin{equation}
\label{law:coassociate}
\q{
  \qcomul[hlen=2]\qspace[-0.5]\qwire\n
  \qspace[0.5]\qcomul[hlen=2]
}
=\q{
  \qwire\qspace[-0.5]\qcomul[hlen=2]\n
  \qcomul[hlen=2]
}
\end{equation}
```



(7)

Example 4.19: Counitality

```
\begin{equation}
\label{law:counit}
\q{
    \qcounit\qwire\n
    \qcomul[hlen=2]
    }
=\q{\qwire[vlen=2]}
=\q{
    \qwire\qcounit\n
    \qcomul[hlen=2]
}
\end{equation}
```

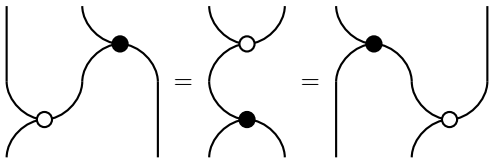


(8)

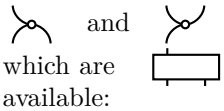
4.4.1 Frobenius Law

Failure Example 4.1: Frobenius Law

```
\[
  \q{
    \qwire\qmul[color=black]\n
    \qcomul\qwire
  }=\q{
    \qcomul\n
    \qmul[color=black]
  }=\q{
    \qmul[color=black]\qwire\n
    \qwire\qcomul
  }
\]
```

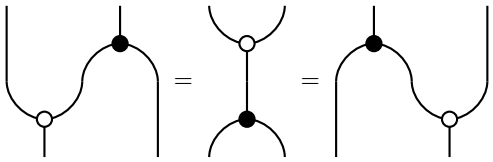


Although it is mentioned as a "Failure Example", in reality, the Frobenius law is accurately described. Rather, since the spacing between all the wires is maintained as integer values, adjustment during vertical composition becomes easier, and in some cases, no adjustment may be necessary. This is merely a matter of preference. The default drawing results of `\qmul` and `\qcomul`, which are

 and , are remnants of the default drawing results of `\qbox[s=2]` and `\qbox[n=2]`, which are and  . For adjustments in such cases, the adjustment key `hlen` is available:

Example 4.20: Frobenius Law

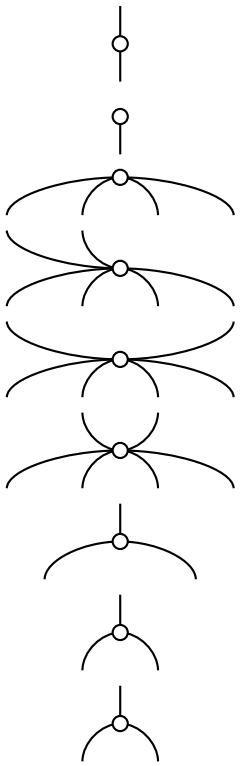
```
\[
  \q{
    \qwire\qmul[hlen=2,color=black]\n
    \qcomul[hlen=2]\qwire
  }=\q{
    \qcomul[hlen=2]\n
    \qmul[hlen=2,color=black]
  }=\q{
    \qmul[hlen=2,color=black]\qwire\n
    \qwire\qcomul[hlen=2]
  }
\]
```



The command `\qspider` is the same as `\qbox`, except that the nodes are circles rather than boxes.

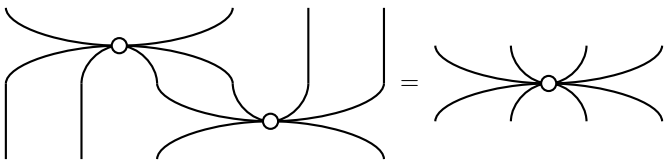
Example 4.21:

```
\[\q{\qspider}\]
\[\q{\qspider[n=0]}\]
\[\q{\qspider[n=0,s=4]}\]
\[\q{\qspider[n=2,s=4]}\]
\[\q{\qspider[N={1,4},s=4]}\]
\[\q{\qspider[N={2,3},s=4]}\]
\[\q{\qspider[p=3,N={2},S={1,3}]}\]
\[\q{\qspider[p=3,N={2},S={1,3},hlen=2]}\]
\[\q{\qspider[n=1,s=2,hlen=2]}\]
```



Example 4.22:

```
\[
  \q{
    \qspider[n=2, s=4, hlen=4]\qwire\qwire\n
    \qwire\qwire\qspider[n=4, s=2, hlen=4]
  }=\q{\qspider[p=4]}\]
```



4.5 Braiding

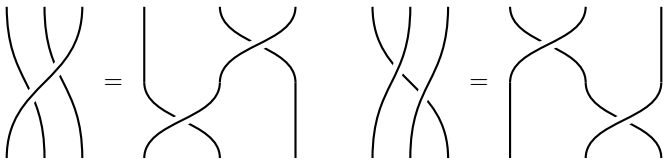
- `\qbraid`: 
- `\qbraidinv`: 

- `\qsym`: 

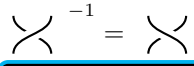
`num L` and `num R` For the commands `\qbraid`, `\qbraidinv`, and `\qsym`, `num L` specifies the number of wires on the left side, and `num R` specifies the number of wires on the right side. For example, specifies `num L=3` and `num R=2` will result in the diagram  . The default value for both is 1.

Example 4.23: Hexagon Equations

```
\begin{equation}
\label{eq:hexagon}
\q
{
  \qbraid[num R=2,,hlen=2,vlen=2]
}=\q
{
  \qwire\qbraid\n
  \qbraid\qwire
}
\qquad
\q
{
  \qbraid[num L=2,,hlen=2,vlen=2]
}
=\q
{
  \qbraid\qwire\n
  \qwire\qbraid
}
\end{equation}
```



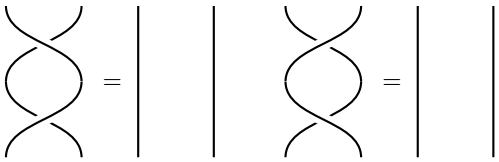
(9)



Example 4.24:

```
\begin{equation}
\label{eq:inverse:braid}
\q
{
  \qbraidinv\n
```

```
  \qbraid
}
=\q
{
  \qwire[vlen=2]\qwire[vlen=2]
}
\qquad
\q
{
  \qbraid\n
  \qbraidinv
}
=\q
{
  \qwire[vlen=2]\qwire[vlen=2]
}
\end{equation}
```

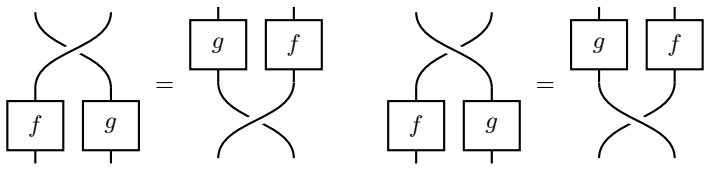


(10)

Example 4.25: Naturality of the Braid (Being a Natural Transformation)

```
\begin{equation}
\label{cond:naturality:braid}
\q
{
  \qbraid\n
  \qbox[name={\(f\)}]\qbox[name={\(g\)}]
}
=\q
{
  \qbox[name={\(g\)}]\qbox[name={\(f\)}]\n
  \qbraid
}
\qquad
\q
{
  \qbraidinv\n
  \qbox[name={\(f\)}]\qbox[name={\(g\)}]
}
=\q
{
  \qbox[name={\(g\)}]\qbox[name={\(f\)}]\n
  \qbraidinv
}
```

```
\end{equation}
```

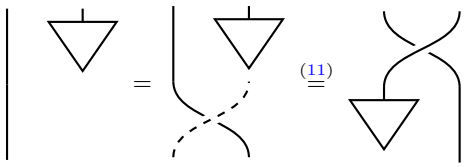


(11)

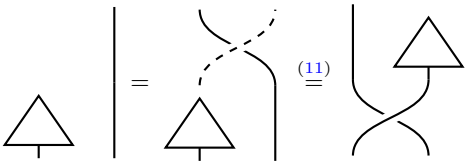
For the commands `\qbraid`, `\qbraidinv`, and `\qasym`, specifying `num R=0` results in a dashed line for the right wire, as shown in  . Similarly, specifying `num L=0` results in a dashed line for the left wire, as shown in  . This is due to representing the unit object as an empty diagrams, i.e., a diagram with 0 wires.

Example 4.26:

```
\begin{gather}
\q{
  \qwire\qstate\n
  \qwire
}
=\q{
  \qwire\qstate\n
  \qbraid[num L=0]
}\stackrel{\text{(\ref{cond:naturality:braid})}}{=}\q{
  \qbraid\n
  \qstate\qwire
}\label{iso:u:state}\\
\q{
  \qunitob\qwire\n
  \qeffect\qwire
}
=\q{
  \qbraid[num L=0]\n
  \qeffect\qwire
}\stackrel{\text{(\ref{cond:naturality:braid})}}{=}\q{
  \qwire\qeffect\n
  \qbraid
}\label{iso:u:effect}
\end{gather}
```



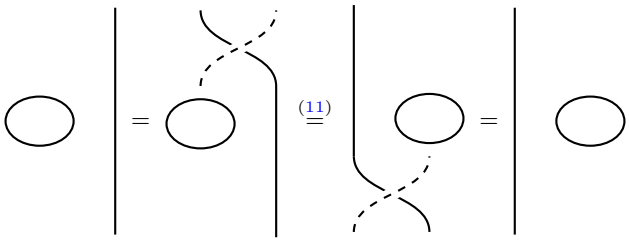
(12)



(13)

Example 4.27:

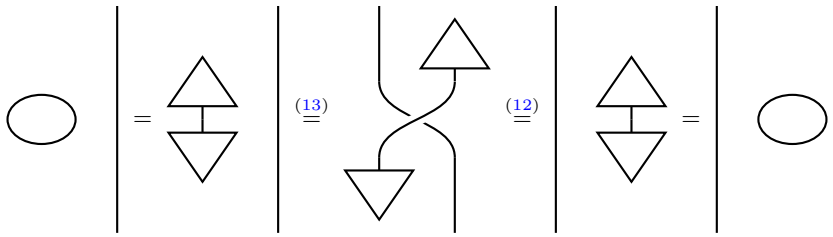
```
\begin{equation}
\label{iso:spatial}
\q{\qscalar\qmv[0,-1]\qwire[vlen=3]}
=\q{\qbraid[num L=0]\n\qscalar\qmv[0,-1]\qwire[vlen=2]}
\stackrel{\text{(\ref{cond:naturality:braid})}}{=}
\q{\qwire[vlen=2]\qscalar\n\qbraid[num L=0]}
=\q{\qscalar\qmv[-2,-1]\qwire[vlen=3]}
\end{equation}
```

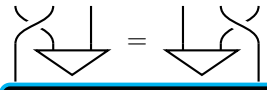


(14)

Example 4.28:

```
\[
  \q{\qscalar\qmv[0,-1]\qwire[vlen=3]}
  =\q{\qeffect\n\qstate\qmv[0,-0.5]\qwire[vlen=3]}
  \stackrel{\text{(\ref{iso:u:effect})}}{=}
  \q{\qwire\qeffect\n\qbraid\n\qstate\qwire}
  \stackrel{\text{(\ref{iso:u:state})}}{=}
  \q{\qeffect\n\qstate\qmv[-2,-0.5]\qwire[vlen=3]}
  =\q{\qscalar\qmv[-2,-1]\qwire[vlen=3]}
\]
```

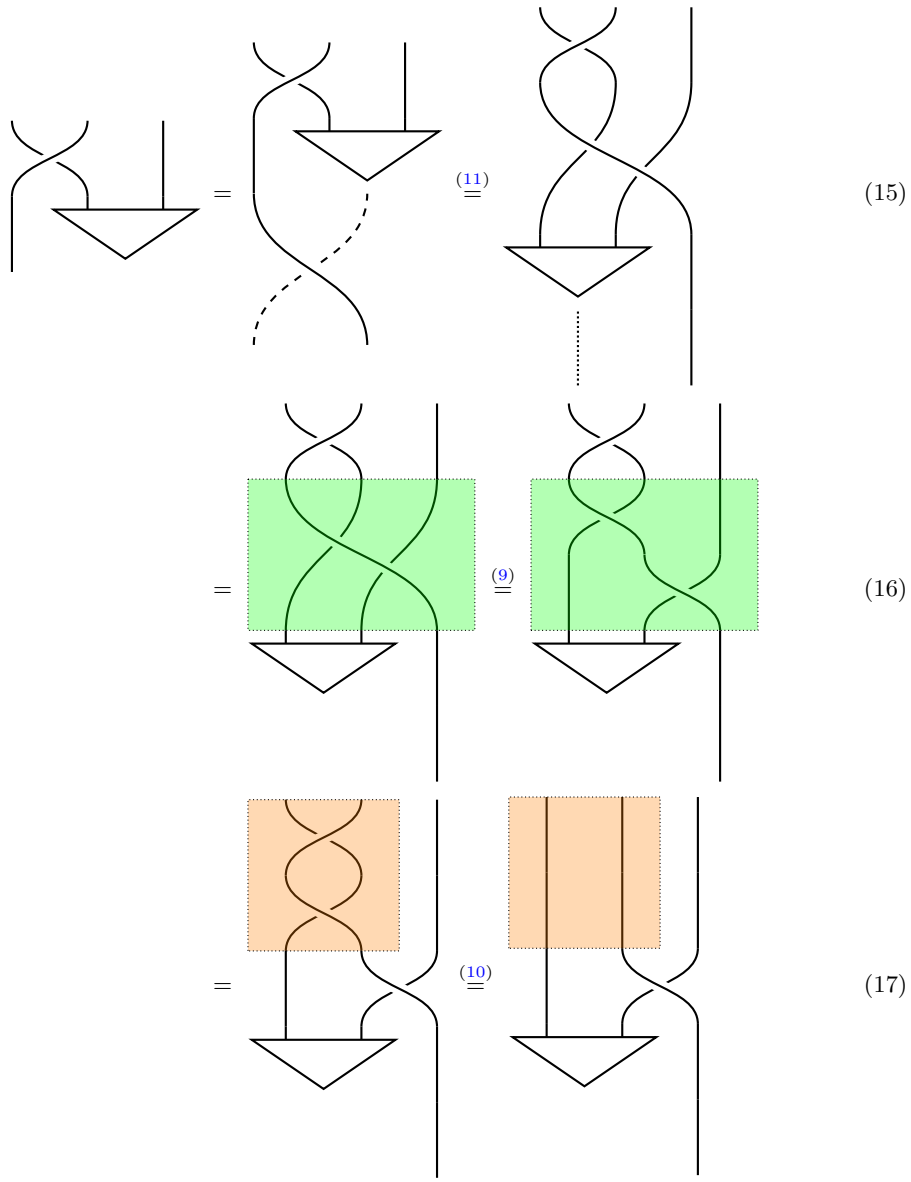




Example 4.29:

```
\begin{align}
\q
{
  \qbraid\qwire\n
  \qwire\qstate[n=2]
}&=\q
{
  \qbraid\qwire\n
  \qwire\qstate[n=2]\n\n
  \qbraidinv[num L=0,vlen=2,hlen=2.5]
}
\stackrel{\text{(\ref{cond:naturality:braid})}}{=}\q
{
  \qbraid\qwire\n\n
  \qbraidinv[num L=2,vlen=2]\n
  \qstate[n=2]\qwire\n
  \qspace[0.5]\qwire[arrowtype={densely dotted}]\qspace[0.5]\qwire
}\\
&=\q
{
  \qbraid\qwire\n\n
  \qbraidinv[num L=2,vlen=2]\n
  \qstate[n=2]\qwire\n
  \qspace[2]\qwire
  \qbb[green][3]
}\stackrel{\text{(\ref{eq:hexagon})}}{=}\q
{
  \qbraid\qwire\n
  \qbraidinv\qwire\n
  \qwire\qbraidinv\n
  \qstate[n=2]\qwire\n
  \qspace[2]\qwire
  \qbb[green][3,...,6]
}\\
```

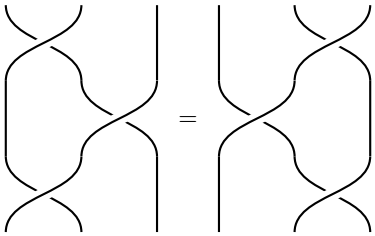
```
&=\q
{
  \qbraid\qwire\n
  \qbraidinv\qwire\n
  \qwire\qbraidinv\n
  \qstate[n=2]\qwire\n
  \qspace[2]\qwire
  \qbb[orange][1,3]
}
\stackrel{\text{(\ref{eq:inverse:braid})}}{=}\q
{
  \qloop[2]{\qloop[3]{\qwire}\n}
  \qwire\qbraidinv\n
  \qstate[n=2]\qwire\n
  \qspace[2]\qwire
  \qbb[orange][1,2,4,5]
}
\end{align}
```



Example 4.30: Yang-Baxer Equation

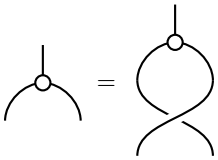
```
\[
  \q{
    \qbraid\qwire\n
    \qwire\qbraid\n
    \qbraid\qwire
  }=\q{
```

```
    \qwire\qbraid\n
    \qbraid\qwire\n
    \qwire\qbraid
  }
\]
```



Example 4.31: Commutativity

```
\begin{equation}
\label{cond:cmt}
\q{\qmul[hlen=2]}=\q{\qmul[hlen=2]\n\qbraid}
\end{equation}
```

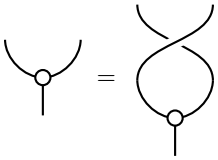


(18)

Example 4.32: Cocommutativity

```
\begin{equation}
\label{cond:ccmt}

  \q{\qcomul[hlen=2]}=\q{\qbraid\n\qcomul[hlen=2]}
\end{equation}
```



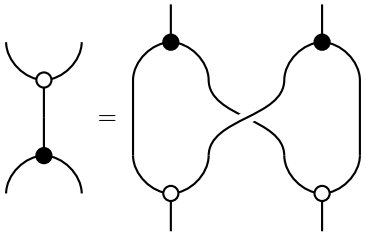
(19)

4.5.1 Bialgebra

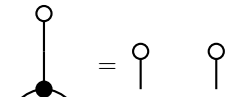
Example 4.33: Bialgebra Law

```
\begin{gather}
\q{\qcomul[hlen=2]\qcirc\qmul[hlen=2, color=black]}
=\q{
\qmul[hlen=2, color=black]\qmul[hlen=2, color=black]\n
\qwire\qbraid\qwire\n\qcomul[hlen=2]\qcomul[hlen=2]
}\\
\q{\qspace[0.5]\qcounit\qcirc\qmul[hlen=2, color=black]}
=\q{\qcounit\qcounit}\\
\q{\qcomul[hlen=2]\qcirc\qspace[0.5]\qunit[color=black]}
=\q{\qunit[color=black]\qunit[color=black]}\\
```

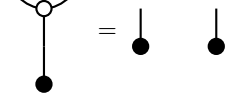
```
\q{\qcounit\n\qunit[color=black]}
=\q{\qunitob}
\end{gather}
```



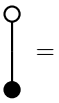
(20)



(21)



(22)

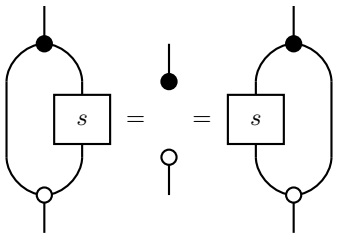


(23)

4.5.2 Hopf Algebra

Example 4.34: Antipode and Hopf Algebra

```
\begin{equation}
\q{
  \qmul[hlen=2, color=black]\n
  \qwire\qbox[name={\(s\)}]\n
  \qcomul[hlen=2]
}
=\q{\qunit[color=black]\qcirc\qcounit}
=\q{
  \qmul[hlen=2, color=black]\n
  \qbox[name={\(s\)}]\qwire\n
  \qcomul[hlen=2]
}
\end{equation}
```

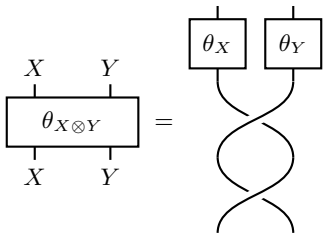


(24)

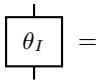
4.5.3 Balanced Structure

Example 4.35: Twist

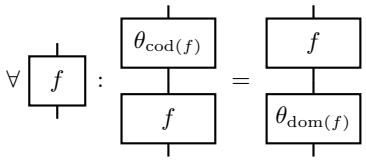
```
\begin{gather}
\q{
  \qbox[name={\(\theta_{X\otimes Y}\)}, p=2]
  \symbolI{\(X\)}\symbolI[2]{\(Y\)}
  \symbolO{\(X\)}\symbolO[2]{\(Y\)}
}
=\q{
  \qbox[name={\(\theta_X\)}]\qbox[name={\(\theta_Y\)}]\n
  \qbraid\n\qbraid
}\label{cond:twist}\\
\q{\qbox[name={\(\theta_I\)}]}=\q{\qunitob}\label{cond:i:twist}\\
\forall\q{\qbox[name={\(f\)}]}\colon
\q{
  \qbox[name={\(\theta_{\operatorname{cod}(f)}\)}, hlen=1.5]\n
  \qbox[name={\(f\)}, hlen=1.5]
}=\q{
  \qbox[name={\(f\)}, hlen=1.5]\n
  \qbox[name={\(\theta_{\operatorname{dom}(f)}\)}, hlen=1.5]
}\label{cond:naturality:twist}
\end{gather}
```



(25)



(26)

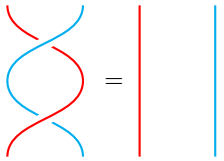


(27)

4.5.4 Color

Example 4.36: Symmetry

```
\begin{equation}
  \label{cond:symmetry}
  \q
  {
    \qbraid[L color=cyan, R color=red]\n
    \qbraid[L color=red, R color=cyan]
  }
  =\q{\qwire[color=red,
  vlen=2]\qwire[color=cyan, vlen=2]}
\end{equation}
```



(28)

Example 4.37:

```
\[\q{\qbraid[L color=red, R color=cyan]}
=\left(\q{\qbraid[L color=cyan, R color=red]}\right)^{-1}
=\q{\qbraidinv[L color=red, R color=cyan]}\]
```

Example 4.38: Symmetry

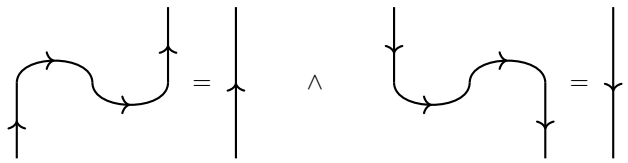
```
\q{\qsym}
```

4.6 Duality

- \qcap: 
- \qcup: 
- \qcaprev: 
- \qcuprev: 

Example 4.39: Snake Equations

```
\begin{equation}
\label{eq:snake}
\q{
  \qcap\qwireup\n
  \qwireup\qcup
}
=\q{
  \qwireup[vlen=2]
}
\qquad\land\qquad
\q{
  \qwiredown\qcap\n
  \qcup\qwiredown
}
=\q{
  \qwiredown[vlen=2]
}
\end{equation}
```

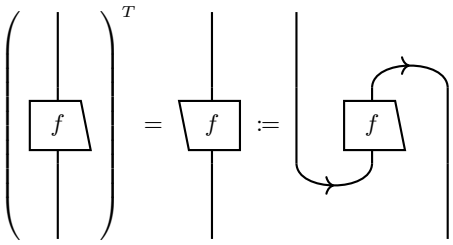


(29)

4.6.1 Transpose Box

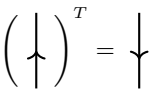
Example 4.40: Transpose

```
\[
  \left(\q{\qwire\n\qasym[name={\(f\)}]\n\qwire}\right)^T
  =\q{
    \qwire\n
    \qtrans[name={\(f\)}]\n
    \qwire
  }\coloneq\q{
    \qwire\qcap\n
    \qwire\qasym[name={\(f\)}]\qwire\n
    \qcup\qwire
  }
\]
```



Example 4.41:

```
\begin{equation}
\label{eq:wirerev}
\left(\q{\qwireup}\right)^T=\q{\qwiredown}
\end{equation}
```

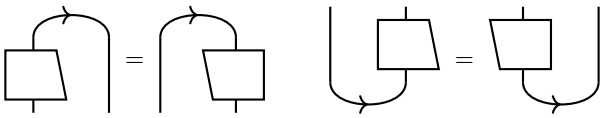


(30)

Example 4.42: Sliding

```
\begin{equation}
\label{eq:sliding}
\q{
  \qcap \n
  \qasym \qwire
}
=\q
{
  \qcap \n
```

```
  \qwire \qtrans
}\qquad
\q
{
  \qwire\qasym \n
  \qcup
}
=\q
{
  \qtrans \qwire \n
  \qcup
}
\end{equation}
```

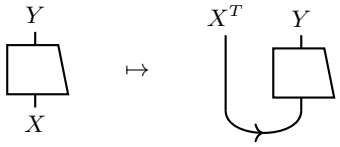


(31)

Example 4.43:

```
\begin{equation}
  \begin{array}{rccc}
    \operatorname{NAME}\colon & \operatorname{Hom}(X,Y)
    & \to & \operatorname{Hom}(I,X^T\otimes Y)\\
    & \rotatebox{90}{\(\in\)} && \rotatebox{90}{\(\in\)}\\
    & \q{
      \qasym
      \symbolI{\(X\)}
      \symbolO{\(Y\)}
    } & \mapsto
    & \q{
      \qwire[label={\(X^T\)}, label at={at end}, label side={above}] \qasym \n
      \qcup
      \symbolO{\(Y\)}[id=2]
    }
  \end{array}
\end{equation}
```

$$\begin{array}{rccc} \text{NAME}\colon & \operatorname{Hom}(X,Y) & \to & \operatorname{Hom}(I,X^T\otimes Y)\\ & \cup & & \cup \end{array}$$

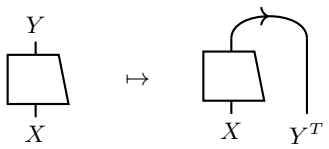


(32)

Example 4.44:

```
\begin{equation}
  \begin{array}{rccc}
    \operatorname{CONAME}\colon & \operatorname{Hom}(X,Y)
    & \to & \operatorname{Hom}(X\otimes Y^T,I)\\
    & \rotatebox{90}{\(\in\)}&&\rotatebox{90}{\(\in\)}\\
    & \q{
      \qasym
      \symbolI{\(X\)}
      \symbolO{\(Y\)}
    } & \mapsto
    & \q{
      \qcap \n
      \qasym \qwire[label={\(Y^T\)}, label at={at start}, label side={below}]
      \symbolI{\(X\)}[id=2]
    }
  \end{array}
\end{equation}
```

$$\text{CONAME:}\quad \underset{\cup}{\mathrm{Hom}(X,Y)} \;\to\; \underset{\cup}{\mathrm{Hom}(X\otimes Y^T,I)} \qquad \underset{\cup}{Y} \;\mapsto\; \underset{\cup}{\;\;\;\;} \tag{33}$$

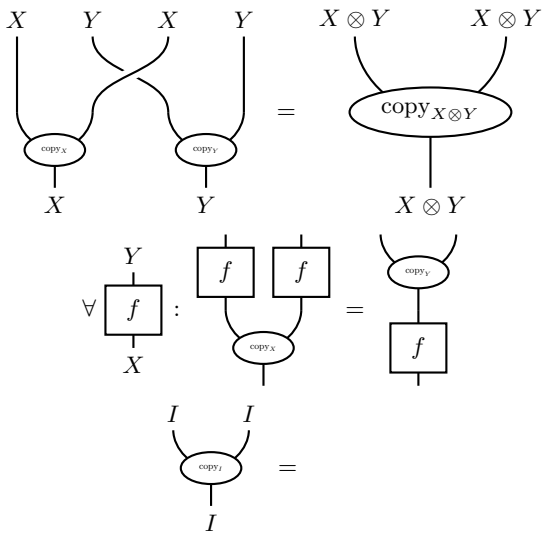


4.6.2 No Cloning Theorem

Example 4.45: Uniform Copying

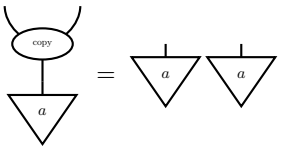
```
\begin{gather}
  \q{
    \qwire\qbraid\qwire\n
    \qcomul[hlen=2, name={\(\operatorname{copy}_X\)}]\qcomul[hlen=2,
  name={\(\operatorname{copy}_Y\)}]
    \symbolI{\(X\)}[id=4]\symbolI{\(Y\)}[id=5]
    \symbolO{\(X\)}[id=1]\symbolO{\(Y\)}[id=2]
    \symbolO[2]{\(X\)}[id=2]\symbolO{\(Y\)}[id=3]
  }=\q{
    \qcomul[hlen=3, vlen=2, name={\(\operatorname{copy}_{X\otimes
  Y}\)}]\symbolI{\(X\otimes Y\)}
    \symbolO{\(X\otimes Y\)}\symbolO[2]{\(X\otimes Y\)}
  }\label{cond:copy}\\
  \forall\q{\qbox[name={\(f\)}]\symbolI{\(X\)}\symbolO{\(Y\)}}\colon
  \q{
    \qbox[name={\(f\)}]\qbox[name={\(f\)}]\n
    \qcomul[hlen=2, name={\(\operatorname{copy}_X\)}]
  }=\q{
```

```
    \qcomul[hlen=2, name={\(\operatorname{copy}_Y\)}]\n
    \qspace[0.5]\qbox[name={\(f\)}]
  }\label{cond:morcopy}\\
  \q{
    \qcomul[hlen=2, name={\(\operatorname{copy}_I\)}]\symbolI{\(I\)}
    \symbolO{\(I\)}\symbolO[2]{\(I\)}
  }=\q{\qunitob}\label{cond:i:copy}
\end{gather}
```



Example 4.46: Copyable State

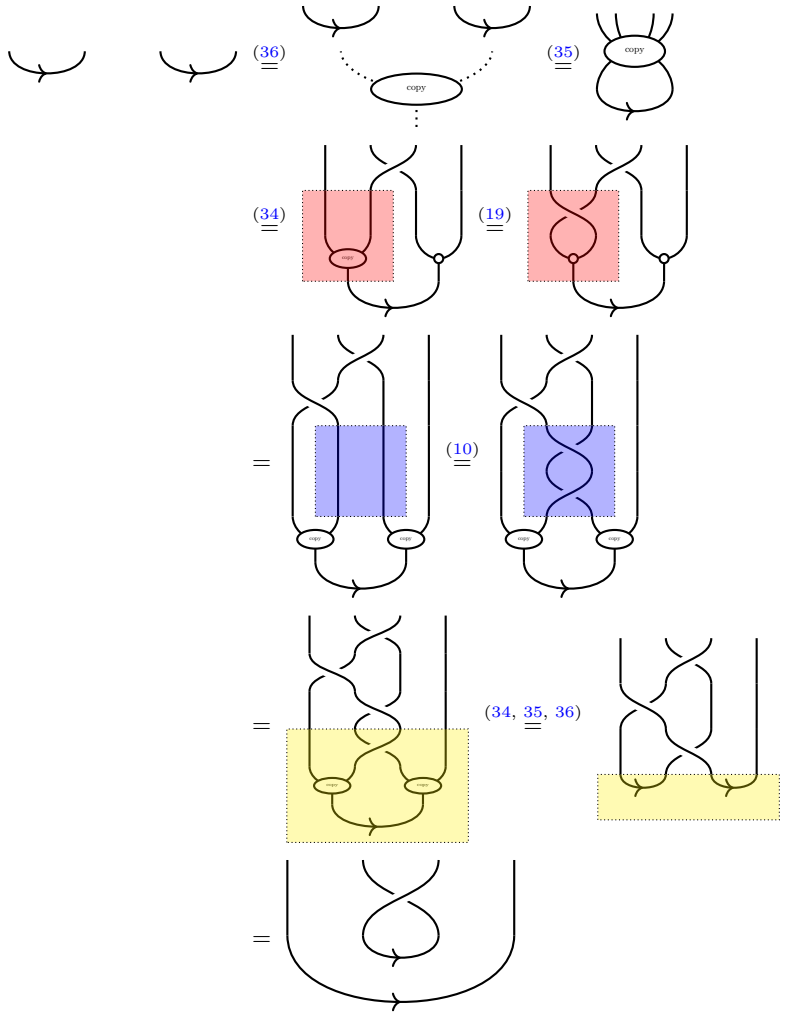
```
\[
  \q{
    \qcomul[hlen=2, name={\(\operatorname{copy}\)}]\n
    \qspace[0.5]\qstate[name={\(a\)}]
  }=\q{
    \qstate[name={\(a\)}]\qstate[name={\(a\)}]
  }
\]
```



Example 4.47:

```
\begin{align*}
  \q{\qcup\qcup}&\stackrel{\text{(\ref{cond:i:copy})}}{=}
  \q{
    \qcup\qcup\n\qmv[0.5,0.4]
    \qcomul[
      dom arrowtype=dotted,
      cod arrowtype=dotted,
      hlen=3,
      name={\(\operatorname{copy}\)}
    ]
  }\stackrel{\text{(\ref{cond:morcopy})}}{=}
  \q{
    \qspider[
      name={\(\operatorname{copy}\)},
      N={1,2,4,5},
      s=2,
      hlen=2
    ]\n\qcup
  }\\
  &\stackrel{\text{(\ref{cond:copy})}}{=}\q[scale=0.6]{
    \qwire\qbraid\qwire\n
    \qloop[4]{\qwire}\n
    \qcomul[hlen=2, name={\(\operatorname{copy}\)}]\qcomul[hlen=2]\n
    \qspace[0.5]\qcup[hlen=3]
    \qbb[red][4,8]
  }\stackrel{\text{(\ref{cond:ccmt})}}{=}\q[scale=0.6]{
    \qwire\qbraid\qwire\n
    \qbraidinv\qwire\qwire\n
    \qcomul[hlen=2]\qcomul[hlen=2]\n
    \qspace[0.5]\qcup[hlen=3]
    \qbb[red][4,7]
  }\\
  &=\q[scale=0.6]{
    \qwire\qbraid\qwire\n
    \qbraidinv\qwire\qwire\n[2]
    \qloop[4]{\qwire[vlen=2]}\n
    \qloop[2]{\qcomul[hlen=2, name={\(\operatorname{copy}\)}]}\n
    \qspace[0.5]\qcup[hlen=3]
    \qbb[blue][8,9]
  }
  \stackrel{\text{(\ref{eq:inverse:braid})}}{=}
  \q[scale=0.6]{
    \qwire\qbraid\qwire\n
    \qbraidinv\qwire\qwire\n
    \qwire\qbraidinv\qwire\n
    \qwire\qbraid\qwire\n
    \qloop[2]{\qcomul[hlen=2, name={\(\operatorname{copy}\)}]}\n
    \qspace[0.5]\qcup[hlen=3]
    \qbb[blue][8,11]
  }\\
```

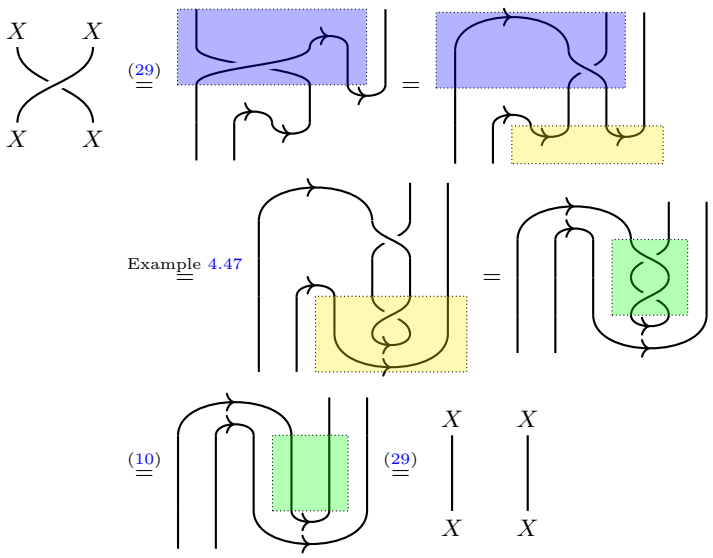
```
  &=\q[xscale=0.6, yscale=0.5]{
    \qwire\qbraid\qwire\n
    \qbraidinv\qwire\qwire\n
    \qwire\qbraidinv\qwire\n
    \qwire\qbraid\qwire\n
    \qloop[2]{\qcomul[hlen=2, name={\(\operatorname{copy}\)}]}\n
    \qspace[0.5]\qcup[hlen=3]
    \qbb[yellow][10,14,15]
  }\stackrel{\text{(\ref{cond:copy}, \ref{cond:morcopy}, \ref{cond:i:copy})}}{=}
  \q[scale=0.6]{
    \qwire\qbraid\qwire\n
    \qbraidinv\qwire\qwire\n
    \qwire\qbraidinv\qwire\n
    \qcup\qcup
    \qbb[yellow][10,11]
  }\\
  &=\q{
    \qwire\qbraid\qwire\n
    \qcup[n=4]
  }
\end{align*}
```



Example 4.48:

```
\begin{align*}
  \q{
    \qbraid
    \symbolI{\(X\)}\symbolO{\(X\)}
    \symbolI[2]{\(X\)}\symbolO[2]{\(X\)}
  }&\stackrel{\text{(\ref{eq:snake})}}{=}\q[scale=0.5]{
    \qwire\qspace[2]\qcap\qwire\n
    \qbraid[hlen=4]\qwire\qwire\n
    \qwire\qcap\qwire\qcup\n
    \qwire\qwire\qcup
    \qbb[blue][1,5]
```

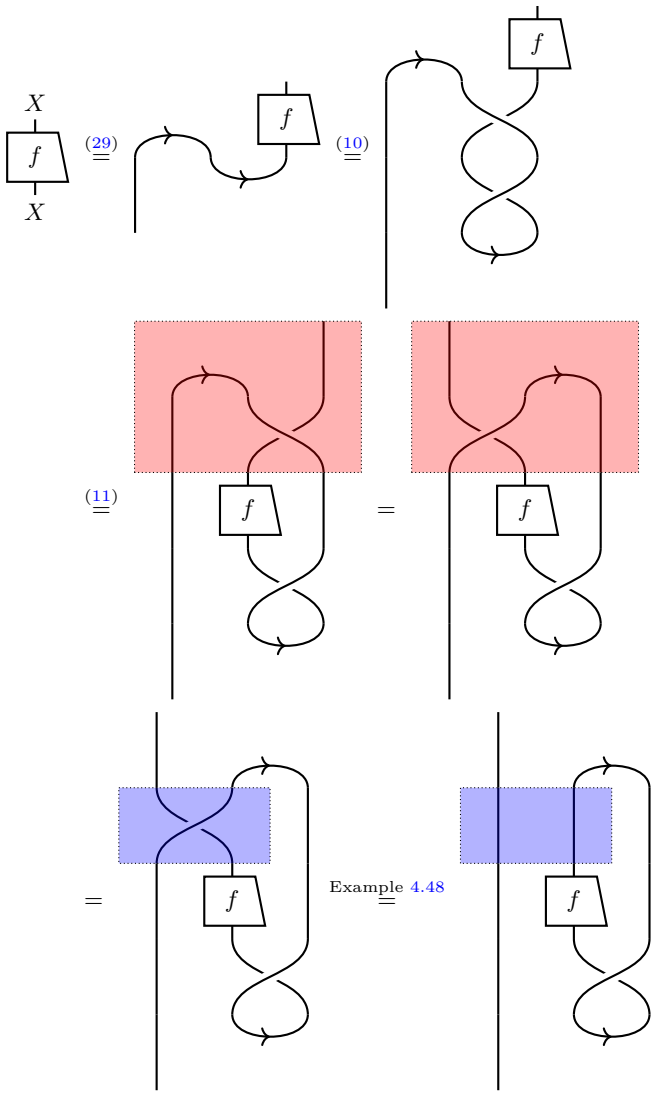
```
  }=\q[scale=0.5]{
    \qcap[hlen=4]\qwire\n
    \qwire\qspace[2]\qbraidinv\n
    \qwire\qcap\qwire\qwire\qwire[vlen=3]\n
    \qwire\qwire\qcup\qcup
    \qbb[blue][1,4]
    \qbb[yellow][12,13]
  }\\
  &\stackrel{\text{Example \ref{len:copycupcup}}}{=}\q[scale=0.5]{
    \qcap[hlen=4]\qwire\n
    \qwire\qspace[2]\qbraidinv\n
    \qwire\qcap\qwire\qwire\qwire[vlen=3]\n
    \qwire\qwire\qwire\qbraid\qwire\n
    \qwire\qwire\qcup[n=4]
    \qbb[yellow][12,14,17]
  }=\q[scale=0.5]{
    \qcap[s=4]\qwire\n
    \qloop[3]{\qwire}\qbraidinv\n
    \qloop[3]{\qwire}\qbraid\qwire[vlen=3]\n
    \qwire\qwire\qcup[n=4]
    \qbb[green][6,10]
  }\\
  &\stackrel{\text{(\ref{eq:inverse:braid})}}{=}\q[scale=0.5]{
    \qcap[s=4]\qwire\qwire\n[2]
    \qloop[6]{\qwire[vlen=2]}\n
    \qwire\qwire\qcup[n=4]
    \qbb[green][7,8]
  }\stackrel{\text{(\ref{eq:snake})}}{=}\q{
    \qwire\qwire
    \symbolI{\(X\)}[id=1]\symbolO{\(X\)}[id=1]
    \symbolI{\(X\)}\symbolO{\(X\)}
  }
\end{align*}
```



Example 4.49: No Cloning Theorem

```
\begin{align*}
  \q{
    \qasym[name={\(f\)}]
    \symbolI{\(X\)}\symbolO{\(X\)}
  }&\stackrel{\text{(\ref{eq:snake})}}{=}
  \q{
    \qcap\qasym[name={\(f\)}]\n
    \qwire\qcup
  }\stackrel{\text{(\ref{eq:inverse:braid})}}{=}\q{
    \qcap\qasym[name={\(f\)}]\n
    \qwire\qbraidinv\n
    \qwire\qbraid\n
    \qwire\qcup
  }\\
  &\stackrel{\text{(\ref{cond:naturality:braid})}}{=}\q{
    \qcap\qwire\n
    \qwire\qbraidinv\n
    \qwire\qasym[name={\(f\)}]\qwire\n
    \qwire\qbraid\n
    \qwire\qcup
    \qbb[red][1,4]
  }=\q{
    \qwire\qcap\n
    \qbraid\qwire\n
    \qwire\qasym[name={\(f\)}]\qwire\n
    \qwire\qbraid\n
    \qwire\qcup
```

```
      \qbb[red][1,4]
    }\\
    &=\q{
      \qwire\qcap\n
      \qbraid\qwire\n
      \qwire\qasym[name={\(f\)}]\qwire\n
      \qwire\qbraid\n
      \qwire\qcup
      \qbb[blue][3]
    }\stackrel{\text{Example \ref{len:copybraid}}}{=}\q{
      \qwire\qcap\n
      \qwire\qwire\qwire\n
      \qwire\qasym[name={\(f\)}]\qwire\n
      \qwire\qbraid\n
      \qwire\qcup
      \qbb[blue][3,4]
    }
  \end{align*}
```

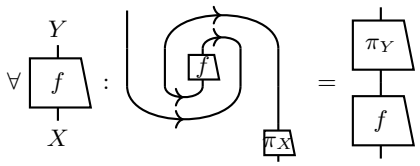


4.7 Pivotal Category

Example 4.50:

```
\begin{equation}
  \label{cond:naturality:pivotal}
  \forall\q{\qasym[name={\(f\)}]\symbolI{\(X\)}\symbolO{\(Y\)}}\colon
  \q[scale=0.5]{
    \qwire\qcap[s=4]\n
    \qwire\qwire\qasym[name={\(f\)}]\qwire\qwire\n
```

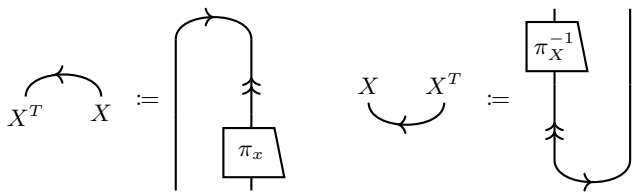
```
    \qcup[n=4]\qwire\qmv[-1,-1]\qasym[name={\(\pi_X\)}]
  }=\q{
      \qasym[name={\(\pi_Y\)}]\n
      \qasym[name={\(f\)}]
  }
\end{equation}
```



(37)

Example 4.51:

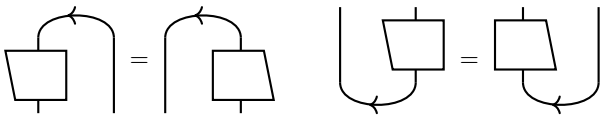
```
\begin{equation}
  \label{def:dualrev}
  \q{
    \qcaprev\n
    \symbolI{\(X^T\)}\symbolI[2]{\(X\)}
  }\coloneq\q
  {
    \qcap \n
    \qwire \qwireupup \n
    \qwire \qasym[name={\(\pi_x\)}] \n
  }\qquad\q{
    \qcuprev
    \symbolO{\(X\)}\symbolO[2]{\(X^T\)}
  }\coloneq\q
  {
    \qasym[name={\(\pi_X^{-1}\)}]\qwire\n
    \qwireupup\qwire\n
    \qcup
  }
\end{equation}
```



(38)

Example 4.52:

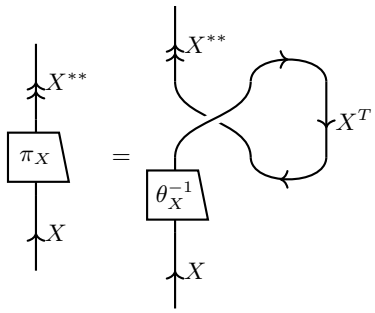
```
\begin{equation}
  \label{eq:sliding:rev}
  \q{
    \qcaprev\n
    \qtrans\qwire\n
  }=\q{
    \qcaprev\n
    \qwire\qasym
  }\qquad\q{
    \qwire\qtrans\n
    \qcuprev
  }=\q{
    \qasym\qwire\n
    \qcuprev
  }
\end{equation}
```



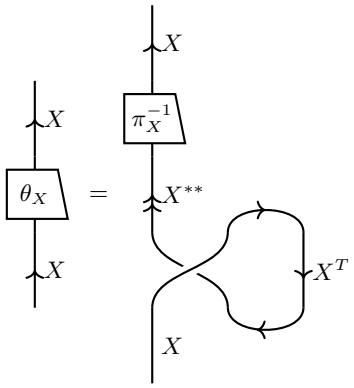
(39)

Example 4.53:

```
\begin{gather}
  \q{
    \qwireuu[label={\(X^{**}\)}]\n
    \qasym[name={\(\pi_X\)}]\n
    \qwireup[label={\(X\)}]
  }=\q{
    \qwireuu[label={\(X^{**}\)}]\qcap\n
    \qbraid\qwired[label={\(X^T\)}]\n
    \qasym[name={\(\theta_X^{-1}\)}]\qcuprev\n
    \qwireup[label={\(X\)}]
  }\label{twist:pivotal}\\
  \q{
    \qwireup[label={\(X\)}]\n
    \qasym[name={\(\theta_X\)}]\n
    \qwireup[label={\(X\)}]
  }=\q{
    \qwireup[label={\(X\)}]\n
    \qasym[name={\(\pi_X^{-1}\)}]\n
    \qwireuu[label={\(X^{**}\)}]\qcap\n
    \qbraid\qwired[label={\(X^T\)}]\n
    \qwire[label={\(X\)}]\qcuprev
  }\label{pivotal:twist}
\end{gather}
```



(40)

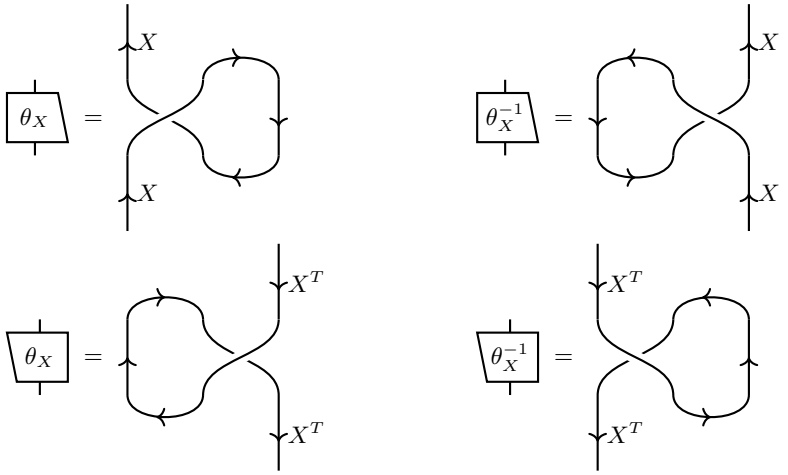


(41)

Example 4.54:

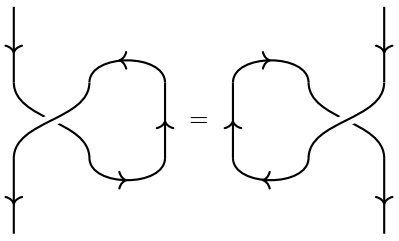
```
\begin{align*}
  \q{\qasym[name={\(\theta_X\)}]}
  &=\q{
    \qwireu[label={\(X\)}]\qcap\n
    \qbraid\qwired\n
    \qwireu[label={\(X\)}]\qcuprev
  }&\q{\qasym[name={\(\theta_X^{-1}\)}]}
  &=\q{
    \qcaprev\qwireu[label={\(X\)}]\n
    \qwired\qbraidinv\n
    \qcup\qwireu[label={\(X\)}]
  }\\
  \q{\qtrans[name={\(\theta_X\)}]}
  &=\q{
    \qcap\qwired[label={\(X^T\)}]\n
    \qwireu\qbraid\n
    \qcuprev\qwired[label={\(X^T\)}]
  }&\q{\qtrans[name={\(\theta_X^{-1}\)}]}
  &=\q{
    \qwired[label={\(X^T\)}]\qcaprev\n
    \qbraidinv\qwireu\n
    \qwired[label={\(X^T\)}]\qcup
```

```
  }
\end{align*}
```



Example 4.55: Ribbon Category

```
\begin{equation}
  \label{def:ribboncategory}
  \q{
    \qwired\qcaprev\n
    \qbraid\qwireu\n
    \qwired\qcup
  }=\q{
    \qcap\qwired\n
    \qwireu\qbraid\n
    \qcuprev\qwired
  }
\end{equation}
```

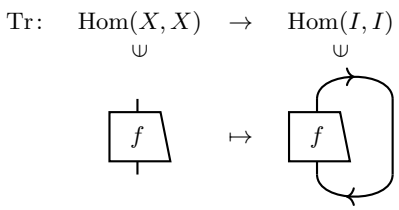


(42)

4.7.1 Trace

Example 4.56: Trace

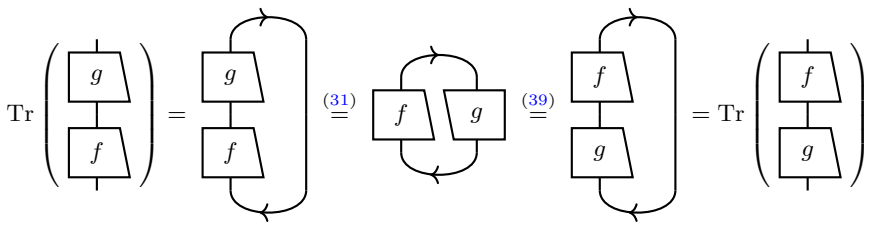
```
\[
  \begin{array}{rccc}
    \operatorname{Tr}\colon & \operatorname{Hom}(X,X) & \to
  &\operatorname{Hom}(I,I)\\
    &\rotatebox{90}{\(\in\)}&&\rotatebox{90}{\(\in\)}\\
    &\q{
      \qasym[name={\(f\)}]
    } & \mapsto & \q{
      \qcap\n
      \qasym[name={\(f\)}]\qwire\n
      \qcuprev
    }
  \end{array}
\]
```



Example 4.57: Commutativity of the Trace

```
\begin{equation}
  \label{cmt:trace}
  \operatorname{Tr}\left(\q{
    \qasym[name={\(g\)}]\n
    \qasym[name={\(f\)}]
    }\right)
  =\q{
    \qcap\n
    \qasym[name={\(g\)}]\qwire\n
    \qasym[name={\(f\)}]\qwire\n
    \qcuprev
    }
  \stackrel{\text{(\ref{eq:sliding})}}{=}\q{
    \qcap\n
    \qasym[name={\(f\)}]\qtrans[name={\(g\)}]\n
    \qcuprev
  }\stackrel{\text{(\ref{eq:sliding:rev})}}{=}\q{
    \qcap\n
    \qasym[name={\(f\)}]\qwire\n
    \qasym[name={\(g\)}]\qwire\n
    \qcuprev
```

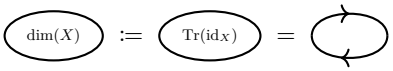
```
  }=\operatorname{Tr}\left(\q{
    \qasym[name={\(f\)}]\n
    \qasym[name={\(g\)}]
    }\right)
\end{equation}
```



(43)

Example 4.58: Dimension

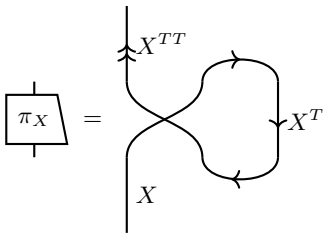
```
\[
  \q{\qscalar[name={\(\operatorname{dim}(X)\)}]}
  \coloneq\q{\qscalar[name={\(\operatorname{Tr}(\operatorname{id}_X)\)}]}
  =\q{\qcap\n\qcuprev}
\]
```



4.7.2 Compact Category

Example 4.59: Compact Category

```
\begin{equation}
  \label{def:compact}
  \q{\qasym[name={\(\pi_X\)}]}=\q{
    \qwireuu[label={\(X^{TT}\)}]\qcap\n
    \qsym\qwired[label={\(X^T\)}]\n
    \qwire[label={\(X\)}]\qcuprev
  }
\end{equation}
```

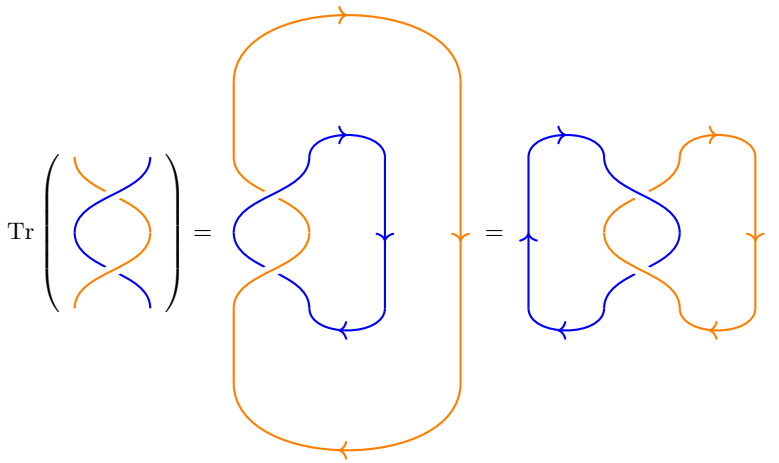


(44)

4.7.3 Color

Example 4.60:

```
\begin{equation}
  \label{eq:tr:hopf}
  \operatorname{Tr}\left(
    \q{
      \qbraid[L color=blue, R color=orange]\n
      \qbraid[L color=orange, R color=blue]}
  \right)
  =\q{
    \qcap[hlen=4, frame color=orange]\n
    \qwire[color=orange]\qcap[frame color=blue]\n
    \qbraid[L color=blue, R color=orange]\n
    \qbraid[L color=orange, R color=blue
    ]\qwiredown[vlen=2, color=blue]\n
    \qwire[color=orange]
    \qcuprev[frame color=blue]
    \qwiredown[vlen=4, color=orange]\n
    \qcuprev[hlen=4, frame color=orange]
  }=\q{
    \qcap[frame color=blue]
    \qcap[frame color=orange]\n
    \qspace
    \qbraidinv[L color=orange, R color=blue]\n
    \qwireup[vlen=2, color=blue]
    \qbraidinv[L color=blue, R color=orange]
    \qwiredown[vlen=2, color=orange]\n
    \qcuprev[frame color=blue]
    \qcuprev[frame color=orange]
  }
\end{equation}
```



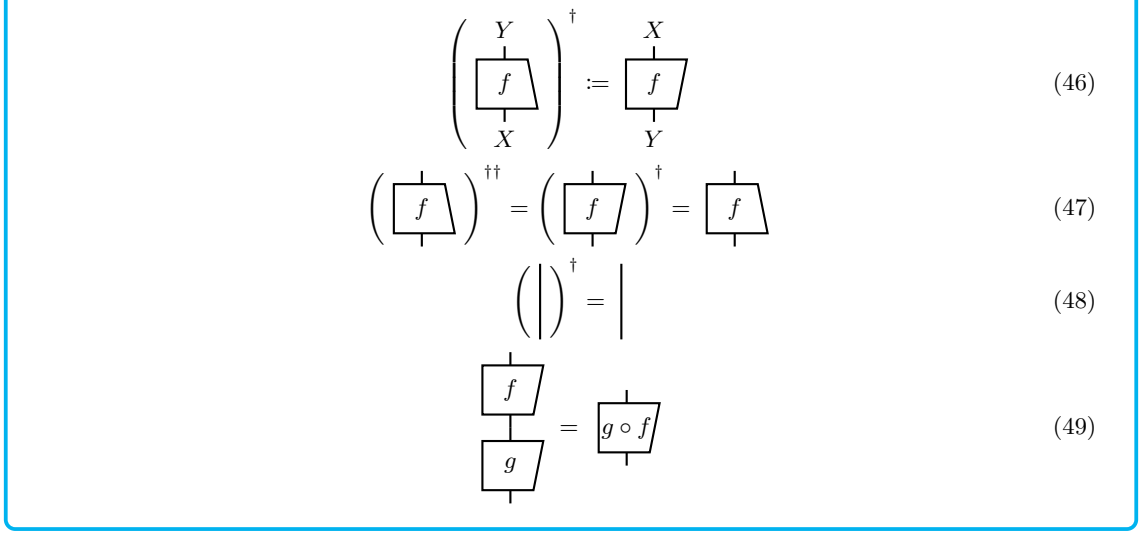
(45)

4.8 Dagger

4.8.1 Adjoint Box

Example 4.61: Adjoint

```
\begin{gather}
  \left(
    \q{
      \qasym[name={\(f\)}]
      \symbolI{\(X\)}\symbolO{\(Y\)}
    }
  \right)^{\dagger}
  \coloneq
  \q{
    \qadj[name={\(f\)}]
    \symbolI{\(Y\)}\symbolO{\(X\)}
  }\\
  \left(\q{\qasym[name={\(f\)}]}\right)^{\dagger\dagger}
  =\left(\q{\qadj[name={\(f\)}]}\right)^{\dagger}
  =\q{\qasym[name={\(f\)}]}\\
  \left(\q{\qwire}\right)^{\dagger}=\q{\qwire}\\
  \q{
    \qadj[name={\(f\)}]\n
    \qadj[name={\(g\)}]
  }=\q{
    \qadj[name={\(g\circ f\)}]
  }
\end{gather}
```



(46)

(47)

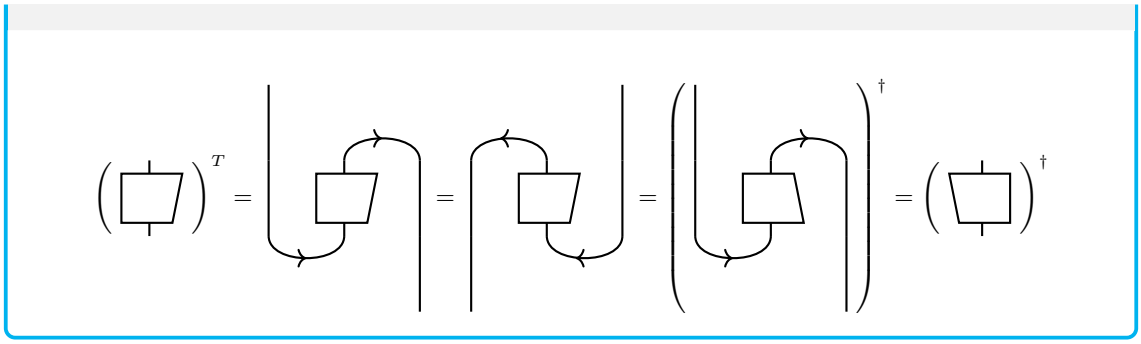
(48)

(49)

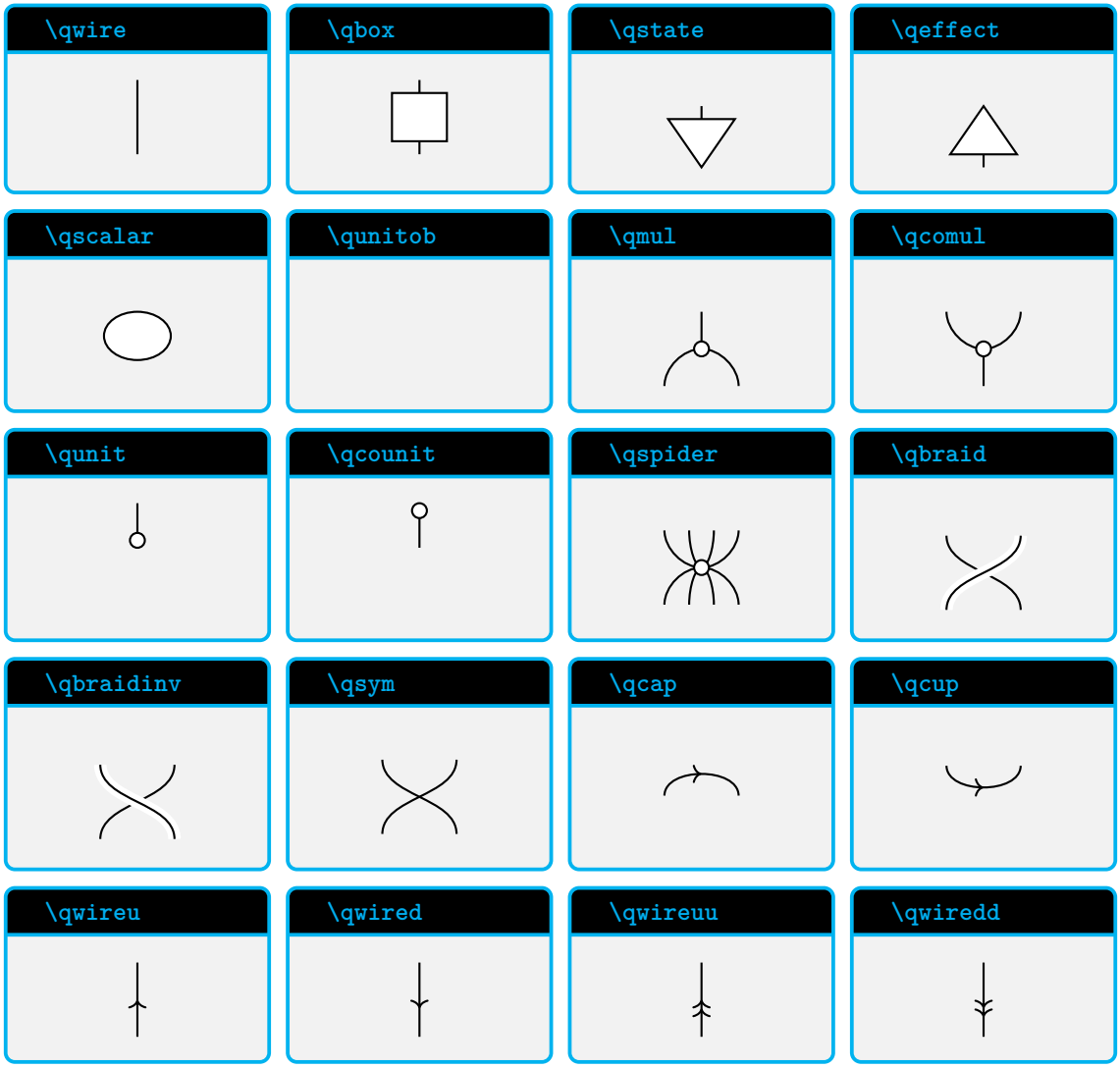
4.8.2 Conjugate Box

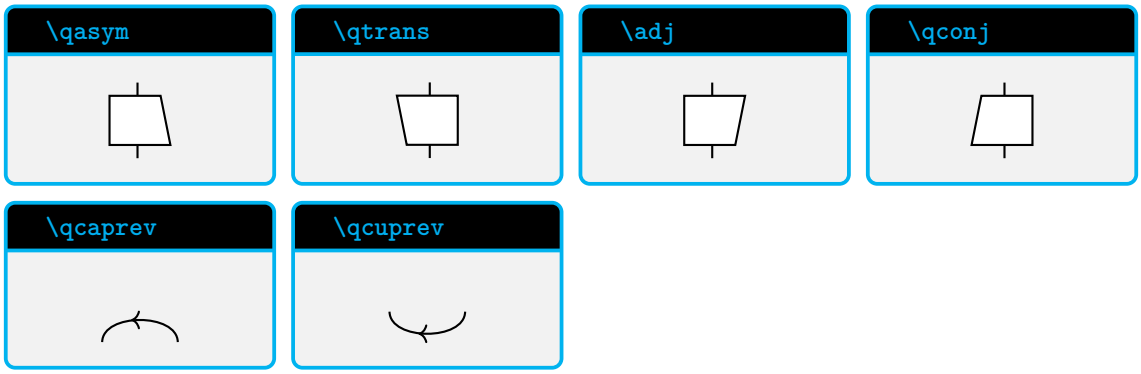
Example 4.62:

```
\begin{align*}
  \left(\q{\qadj}\right)^T
  &=\q{
    \qwire\qcap\n
    \qwire\qadj\qwire\n
    \qcup\qwire
  }=\q{
    \qcaprev\qwire\n
    \qwire\qadj\qwire\n
    \qwire\qcuprev
  }=\left(
    \q{
      \qwire\qcap\n
      \qwire\qasym\qwire\n
      \qcup\qwire
    }
  \right)^{\dagger}=\left(\q{\qtrans}\right)^{\dagger}
\end{align*}
```



5 List of Primitive Diagrams





References

[1] Till Tantau, *The TikZ and PGF Packages Manual for Version 3.1.10*, Institut für Theoretische Informatik, Universität zu Lübeck, n.d., https://github.com/pgf-tikz/pgf (retrieved March 22, 2025).

[2] Anonymous, *Genius Sorting Algorithm*, 4chan Programming Board, January 20, 2011, https://dis.4chan.org/read/prog/1295544154 (accessed January 20, 2011; link currently inactive).

[3] gfx, *Joushiki wo kutsugaesu so-to arugorizumu! Sono na mo "sleep sort"!*, Hatena Diary, May 19, 2011, https://gfx.hatenadiary.org/entry/20110519/1305810786.

[4] LiteratePrograms, *Turing Machine Simulator in LaTeX*, https://literateprograms.org/turing_machine_simulator__latex_.html, n.d., retrieved March 22, 2025.

[5] Overleaf, *LaTeX is More Powerful than You Think: Computing the Fibonacci Numbers and Turing Completeness*, https://www.overleaf.com/learn/latex/Articles/LaTeX_is_More_Powerful_than_you_Think_-_Computing_the_Fibonacci_Numbers_and_Turing_Completeness, n.d., retrieved March 22, 2025.

Changelog

1.1.1 Ninna Ryota (2025-04-14)

Added

- Color annotation for wires and tangles

  The `frame color` attribute is now supported for CAP and CUP structures, allowing wires to express object identity through color. Additionally, BRAIDING now supports separate `L color` and `R color` options for independently coloring the left and right wires.

Fixed

- Improved wire rendering

  The rendering of curves has been refined to produce smoother and more aesthetically coherent diagrams.

1.1.0 Ninna Ryota (2025-04-03)

Changed

- Argument structure revised for symbol commands
  The following commands were updated to support key-value-style optional arguments instead of mandatory positional arguments:
  - `\symbolI`
  - `\symbols`
  - `\symbolS`
  - `\symbolO`
  - `\symboln`
  - `\symbolN`

  Previous syntax `\symbolI[i]{ID}{text}[position options]`
  New syntax `\symbolI[i]{text}[id=ID, position options]`

  With this revision, the `ID` parameter is now optional. In simple cases, `\symbolI{text}` suffices. This change enables more concise and readable definitions.

Added

- English documentation
  An English version of the user guide has been added.

1.0.0 Ninna Ryota (2025-04-01) — Initial release

License and Copyright