alchemist

v0.1.0 MIT

A package to render skeletal formulas using cetz

Robotechnic <@Robotechnic>

https://github.com/Robotechnic/alchemist

Alchemist is a package used to draw chemical structures with skeletal formulas using Cetz. It is heavily inspired by the Chemfig package for LaTeX. This package is meant to be easy to use and customizable. It can also be used alongside the cetz package to draw more complex structures.

Table of contents

I. Usage

I.1. Initializing drawing environment 2

I.2. Configuration 2

I.3. Available commands 3

I.3.1. Molecule function 3

I.3.2. Branch and cycles 3

II. Drawing molecules

II.1. Basic drawing 5

II.2. Integration with cetz 5

III. Index

Part I.
Usage

To start using Alchemist, just import the package in your document:

```
#import "@preview/alchemist:0.1.0": *
```

I.1. Initializing drawing environment

To start drawing molecules, you first need to initialise the drawing environment. This is done by calling the `#skeletize()` function.

```
#skeletize({
  ...
})
```

The main argument is a block of code that contains the drawing instructions. The block can also contain any cetz code to draw more complex structures, see Section II.2.

```
#skeletize(⟨debug⟩: false, ⟨background⟩: none, ⟨config⟩: (:), ⟨body⟩)
```

Argument

⟨debug⟩ `bool`

Display bounding boxes of the objects in the drawing environment.

Argument

⟨background⟩ `color` | `none`

Background color of the drawing environment

Argument

⟨config⟩ `dictionary`

Configuration of the drawing environment. See Section I.2.

Argument

⟨body⟩ `drawable`

The module to draw or any cetz drawable object.

I.2. Configuration

Th configuration dictionary that you can pass to skeletize defines a set of default values for a lot of parameters in alchemist.

Argument

⟨atom-sep⟩: 3em `length`

It defines the distance between each atom center. It is overridden by the `atom-sep` argument of link

Argument

⟨angle-increment⟩: 45deg `angle`

It defines the angle added by each increment of the `angle` argument of link

Argument

⟨base-angle⟩: 0deg `angle`

Default angle at which the link with no angle defined will be.

I.3. Available commands

I.3.1. Molecule function

#molecule(⟨name⟩: none, ⟨links⟩: "(:)", ⟨mol⟩) → drawable

Build a molecule group based on mol Each molecule is represented as an optional count followed by a molecule name starting by a capital letter followed by an optional indice

```
#skeletize({
  molecule("H_2O")
})
```

H_2O

```
#skeletize({
  molecule("2Aa_7 3Bb")
})
```

$2\mathrm{Aa}_7 3\mathrm{Bb}$

Argument

⟨name⟩: none `str`

The name of the molecule. It is used as the cetz name of the molecule and to link other molecules to it.

Argument

⟨links⟩: "(:)" `dictionary`

The links between this molecule and the previous ones. The key is the name of the molecule and the value is the link you want to draw between the two molecules.

Note that the length and angle arguments are ignored

Argument

⟨mol⟩ `str`

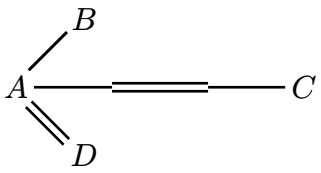
The string representing the molecule

I.3.2. Branch and cycles

#branch(⟨body⟩)

Create a branch from the current molecule, the first element of the branch has to be a link

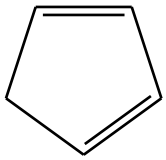
```
#skeletize({
  molecule("A")
  branch({
    single(angle:1)
    molecule("B")
  })
  branch({
    double(angle: -1)
    molecule("D")
  })
  single()
  double()
  single()
  molecule("C")
})
```



#cycle(..⟨args⟩)

Create a regular cycle of molecules

```
#skeletize({
  cycle(5, {
    single()
    double()
    single()
    double()
    single()
  })
})
```



Part II.
Drawing molecules

II.1. Basic drawing

II.2. Integration with cetz

Part III.
Index

B

#branch 3

C

#cycle 4

M

#molecule 3

S

#skeletize 2