"Monocle" library usage by example

Osman Bineev

Start GHCi and load library:

```
Prelude> :l Monocle
...
Ok, modules loaded: Monocle, Monocle.Rules, Monocle.Tex, Monocle.Markup, Monocle.Core, Monocle.Utils.
```

Or just import if you installed it before:

```
Prelude> import Monocle
```

Create objects A and B:

```
>    let a = object "A"; b = object "B"
```

Create morphisms $f : A \to B$ and $g : B \otimes A \to B \otimes A$:

```
>    let f = arrow "f" a b
>    let g = arrow "g" (b \* a) (b \* a)
```

Create the arrow $h = g \circ (f \otimes id_A)$:

```
>    let h = g \. f \* a
```

Now we can print the morphism formula in LaTeX:

```
>    ptex h
$g\circ (f\otimes  id_{A})$
```

Result: $g \circ (f \otimes id_A)$.
Or more detailed information:

```
>    pdoc h
\textsf{morphism }$g\circ (f\otimes  id_{A}): A\otimes A\to B\otimes A$
\textsf{ where }
\begin{itemize}
\item
\textsf{morphism }$f: A\to B$

\item
\textsf{morphism }$g: B\otimes A\to B\otimes A$
\end{itemize}
```

Result:

morphism $g \circ (f \otimes id_A) : A \otimes A \to B \otimes A$ where

- morphism $f : A \to B$
- morphism $g : B \otimes A \to B \otimes A$

Next, create braid β and unbraid β^{-1}:

```
>    let br = braid a b; ub = unbraid b a
```

Show them in LaTeX:

```
>    pdoc br
>    pdoc ub
```

morphism $\beta_{A,B} : A \otimes B \to B \otimes A$

morphism $\beta^{-1}_{B,A} : B \otimes A \to A \otimes B$

Show the unbraid rule:

```
>    pdoc braid'rule'Iso'Left
```

rule $(\beta^{-1}_{B,A} \circ \beta_{A,B}) \equiv (id_A \otimes id_B)$ where

- morphism $\beta^{-1}_{B,A} \circ \beta_{A,B} : A \otimes B \to A \otimes B$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $\beta^{-1} : B \otimes A \to A \otimes B$
- morphism $id_A \otimes id_B : A \otimes B \to A \otimes B$ where

apply it to $\beta^{-1} \circ \beta$:

```
>    let r = apply braid'rule'Iso'Left (ub \. br)
```

Result:

```
>    ptex r
```

$id_A \otimes id_B$

How to apply a rule "deeply inside" the formula? Let's create morphism $h : A \otimes B \to A \otimes B$. Suppose we need to apply the unbraid rule to $h_1 = (h \circ \beta^{-1} \circ \beta \circ h) \otimes h$:

```
>    let h = arrow "h" (a \* b) (a \* b)
>    let h1 = (h \. ub \. br \. h) \* h
```

First we should markup h_1:

```
>    let h2 = markup h1
>    ptex h2
```

$((h \circ (\beta_{A,B})_{lab:1} \circ (\beta^{-1}_{B,A})_{lab:2} \circ h)_{lab:3} \otimes h)_{lab:4}$

In the subterm labelled as $lab : 3$ we should select $\beta^{-1} \circ \beta$. We use the *choose* function with three arguments: the name of the new label, positions of the first and the last arguments of the subformula. The $modifLab$ function allows to do it by existing label $lab : 3$.

```
>    let h3 = modifLab "lab:3" h2 $ choose "lab" 2 3
>    ptex h3
```

$((h \circ ((\beta_{A,B})_{lab:1} \circ (\beta^{-1}_{B,A})_{lab:2})_{lab} \circ h)_{lab:3} \otimes h)_{lab:4}$

Now the unbraid rule can be applpied to the existing label lab:

```
>    ptex $ modif "lab" h3 $ apply braid'rule'Iso'Left
```

$(h \circ h) \otimes h$

Documentation printed for all rules of Monocle-0.0.4:
zigzag'rule'Left:

rule $(\epsilon_{A,B} \otimes id_A) \circ (id_A \otimes \eta_{A,B}) \equiv id_A$ where

- morphism $(\epsilon_{A,B} \otimes id_A) \circ (id_A \otimes \eta_{A,B}) : A \to A$ where
  - morphism $\epsilon : A \otimes B \to I$

  - morphism $\eta : I \to B \otimes A$

zigzag'rule'Right:

rule $(id_B \otimes \epsilon_{A,B}) \circ (\eta_{A,B} \otimes id_B) \equiv id_B$ where

- morphism $(id_B \otimes \epsilon_{A,B}) \circ (\eta_{A,B} \otimes id_B) : B \to B$ where
  - morphism $\epsilon : A \otimes B \to I$
  - morphism $\eta : I \to B \otimes A$

braid'rule'Iso'Left:

rule $(\beta^{-1}_{B,A} \circ \beta_{A,B}) \equiv (id_A \otimes id_B)$ where

- morphism $(\beta^{-1}_{B,A} \circ \beta_{A,B}) : A \otimes B \to A \otimes B$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $\beta^{-1} : B \otimes A \to A \otimes B$
- morphism $(id_A \otimes id_B) : A \otimes B \to A \otimes B$

braid'rule'Iso'Right:

rule $(\beta_{B,A} \circ \beta^{-1}_{A,B}) \equiv (id_A \otimes id_B)$ where

- morphism $(\beta_{B,A} \circ \beta^{-1}_{A,B}) : A \otimes B \to A \otimes B$ where
  - morphism $\beta : B \otimes A \to A \otimes B$
  - morphism $\beta^{-1} : A \otimes B \to B \otimes A$
- morphism $(id_A \otimes id_B) : A \otimes B \to A \otimes B$

braid'rule'Nat'Left:

rule $(\beta_{A,B} \circ (f \otimes id_B)) \equiv ((id_B \otimes f) \circ \beta_{A,B})$ where

- morphism $(\beta_{A,B} \circ (f \otimes id_B)) : A \otimes B \to B \otimes A$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $f : A \to A$
- morphism $((id_B \otimes f) \circ \beta_{A,B}) : A \otimes B \to B \otimes A$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $f : A \to A$

braid'rule'Nat'Right:

rule $(\beta_{A,B} \circ (id_A \otimes f)) \equiv ((f \otimes id_A) \circ \beta_{A,B})$ where

- morphism $(\beta_{A,B} \circ (id_A \otimes f)) : A \otimes B \to B \otimes A$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $f : B \to B$
- morphism $((f \otimes id_A) \circ \beta_{A,B}) : A \otimes B \to B \otimes A$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $f : B \to B$

braid'rule'Hex'Braid:

rule $(id_B \otimes \beta_{A,C}) \circ (\beta_{A,B} \otimes id_C) \equiv \beta_{A,(B \otimes C)}$ where

- morphism $(id_B \otimes \beta_{A,C}) \circ (\beta_{A,B} \otimes id_C) : A \otimes B \otimes C \to B \otimes C \otimes A$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $\beta : A \otimes C \to C \otimes A$

- morphism $\beta_{A,(B \otimes C)} : A \otimes B \otimes C \to B \otimes C \otimes A$

braid'rule'Hex'Unbraid:

rule $(id_B \otimes \beta^{-1}_{A,C}) \circ (\beta^{-1}_{A,B} \otimes id_C) \equiv \beta^{-1}_{A,(B \otimes C)}$ where

- morphism $(id_B \otimes \beta^{-1}_{A,C}) \circ (\beta^{-1}_{A,B} \otimes id_C) : A \otimes B \otimes C \to B \otimes C \otimes A$ where
  - morphism $\beta^{-1} : A \otimes B \to B \otimes A$
  - morphism $\beta^{-1} : A \otimes C \to C \otimes A$
- morphism $\beta^{-1}_{A,(B \otimes C)} : A \otimes B \otimes C \to B \otimes C \otimes A$

cross'rule

rule $\beta_{A,B} \equiv \beta^{-1}_{A,B}$ where

- morphism $\beta_{A,B} : A \otimes B \to B \otimes A$
- morphism $\beta^{-1}_{A,B} : A \otimes B \to B \otimes A$

twist'rule'Iso'Left:

rule $(\theta^{-1}_A \circ \theta_A) \equiv (id_A \otimes id_B)$ where

- morphism $(\theta^{-1}_A \circ \theta_A) : A \to A$ where
  - morphism $\theta : A \to A$
  - morphism $\theta^{-1} : A \to A$
- morphism $(id_A \otimes id_B) : A \otimes B \to A \otimes B$

twist'rule'Iso'Right:

rule $(\theta_A \circ \theta^{-1}_A) \equiv id_A$ where

- morphism $(\theta_A \circ \theta^{-1}_A) : A \to A$ where
  - morphism $\theta : A \to A$
  - morphism $\theta^{-1} : A \to A$

twist'rule'Id:

rule $\theta_I \equiv id_I$ where

- morphism $\theta_I : I \to I$

twist'rule'Natural:

rule $(\theta_A \circ f) \equiv (f \circ \theta_A)$ where

- morphism $(\theta_A \circ f) : A \to A$ where
  - morphism $\theta : A \to A$
  - morphism $f : A \to A$
- morphism $(f \circ \theta_A) : A \to A$ where
  - morphism $\theta : A \to A$
  - morphism $f : A \to A$

twist'rule'Braid:

rule $(\beta_{B,A} \circ (\theta_B \otimes \theta_A) \circ \beta_{A,B}) \equiv \theta_{(A \otimes B)}$ where

- morphism $(\beta_{B,A} \circ (\theta_B \otimes \theta_A) \circ \beta_{A,B}) : A \otimes B \to A \otimes B$ where
  - morphism $\beta : A \otimes B \to B \otimes A$
  - morphism $\beta : B \otimes A \to A \otimes B$

- morphism $\theta : A \to A$
- morphism $\theta : B \to B$

- morphism $\theta_{(A \otimes B)} : A \otimes B \to A \otimes B$

dagger'rule'Id:

rule $dagger(id_A) \equiv id_A$ where

- morphism $dagger(id_A) : dagger(A) \to dagger(A)$

dagger'rule'Cofunctor:

rule $dagger(f \circ g) \equiv (dagger(g) \circ dagger(f))$ where

- morphism $dagger(f \circ g) : dagger(C) \to dagger(A)$ where
  - morphism $f : B \to C$
  - morphism $g : A \to B$
- morphism $(dagger(g) \circ dagger(f)) : dagger(C) \to dagger(A)$ where
  - morphism $f : B \to C$
  - morphism $g : A \to B$

qdoc dagger'rule'Inv:

rule $dagger(dagger(f)) \equiv f$ where

- morphism $dagger(dagger(f)) : dagger(dagger(A)) \to dagger(dagger(A))$ where
  - morphism $f : A \to A$
- morphism $f : A \to A$