Consider a monoidal category, we introduce a graphical language[1] where

- objects (and identities) are wires

```
a ────────
```

- morphisms are boxes

```
   ┌─────────┐
a ─┤    f    ├─ b
   └─────────┘
```

- morphism composition wires boxes together

```
   ┌─────────┐      ┌─────────┐                 ┌─────────┐
a ─┤    f    ├──────┤    g    ├─── c  =  a ─────┤   g.f   ├─── c
   └─────────┘      └─────────┘                 └─────────┘
```

- tensor product places objects side by side

```
                     a ────────
a⊗b ────────    =
                     b ────────
```

- tensor product places morphisms side by side

```
   ┌─────────┐               ┌──────────┐
a ─┤    f    ├─ c      a ────┤          ├──── c
   └─────────┘               │          │
                   =         │   f⊗g    │
   ┌─────────┐               │          │
b ─┤    g    ├─ d      b ────┤          ├──── d
   └─────────┘               └──────────┘
```

[1] read left → right

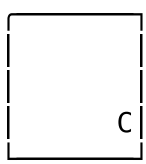
Graphical languages can have two property:

- **soundess**: the defining axioms of category theory are satisfied in the graphical language
- **completeness**: every equation thet holds in the graphical language is a consequence of the axioms

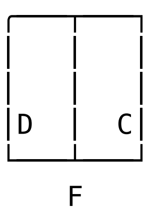
Theorem 0.1. *(Coherence) A well formed equation between two morphisms in the language of category theory follows from the axioms iff it holds in the graphical language (up to isomorphism).*

Any functor category can be made into a monoidal category by taking functor composition as the tensor product, we introduce a graphical language[2] where

- a category[3] is a portion of empty plane

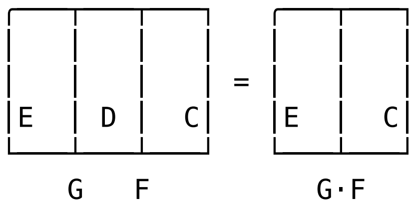


- functors are vertical edges ($F : \mathcal{C} \to \mathcal{D}$)

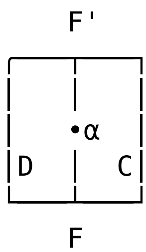


- functor composition places functors side by side ($F : \mathcal{C} \to \mathcal{D}, G : \mathcal{D} \to \mathcal{E}$)

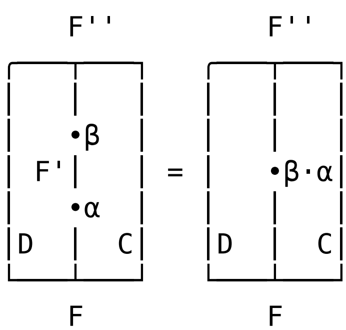
[2] read right → left, bottom → top
[3] and also its identity functor $\mathbf{1} : \mathcal{C} \to \mathcal{C}$



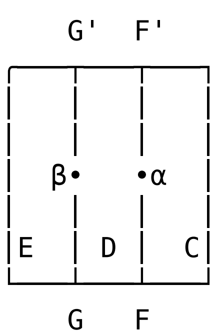
- natural transformations are <u>dots</u> ($\alpha : F \Rightarrow F'$)



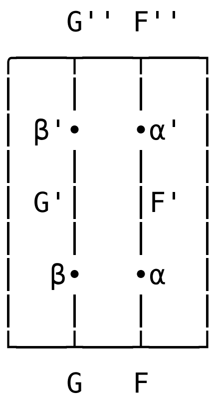
- vertical composition ($\alpha : F \Rightarrow F', \beta : F' \Rightarrow F''$)



- horizontal composition ($\beta \star \alpha$)



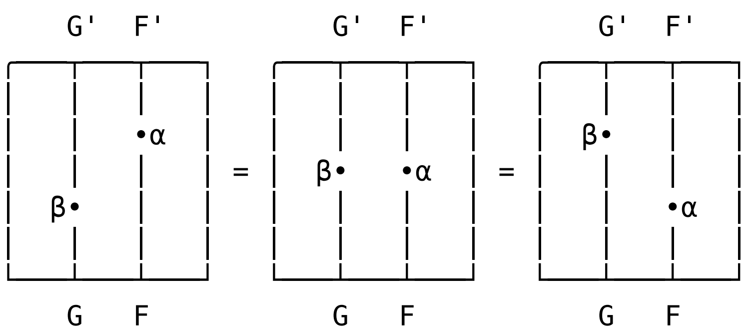
The diagram



is well defined because of the **interchange law**

$$(\beta' \circ \beta) \star (\alpha' \circ \alpha) = (\beta \star \alpha) \circ (\beta' \star \alpha')$$

and substituting identities we get the **sliding equalities**



See [1], [2] and

- Category Theory III 2.1: String Diagrams part 1 - YouTube
- Category Theory III 2.2, String Diagrams part 2 - YouTube
- string diagram in nLab
- TikZiT

[1] D. Marsden, *Category Theory Using String Diagrams*, (2014).
[2] P. Selinger, *A Survey of Graphical Languages for Monoidal Categories*, in *New Structures for Physics* (Springer Berlin Heidelberg, 2010), pp. 289–355.