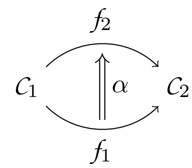
2-Categories

Ross Tate

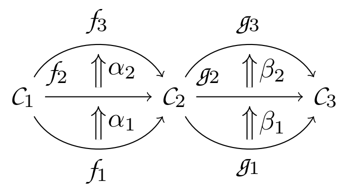
September 23, 2014

Definition (2-Category)**.** A **Cat**-enriched category.

Remark. The objects of a 2-category are called objects or 0-cells. Given two objects C_1 and C_2, the objects of the category $\mathbf{M}(C_1, C_2)$ are called the morphisms or 1-cells of the 2-category. Given two objects C_1 and C_2, the morphisms of the category $\mathbf{M}(C_1, C_2)$ from f_1 to f_2 are called the faces or 2-cells of the 2-category. They are often depicted as in the following, with α being a 2-cell:



Composition of 2-cells within the category $\mathbf{M}(C_1, C_2)$ is called vertical composition. The process of taking 2-cells in $\mathbf{M}(C_1, C_2)$ and 2-cells in $\mathbf{M}(C_2, C_3)$ to 2-cells in $\mathbf{M}(C_1, C_3)$ given by enrichment is called horizontal composition. The fact that this process is a bifunctor guarantees that all the ways to compose the 2-cells in the following diagram produce the same result:



Example. Cat and **CAT** can be made into 2-categories by using natural transformations as the 2-cells.

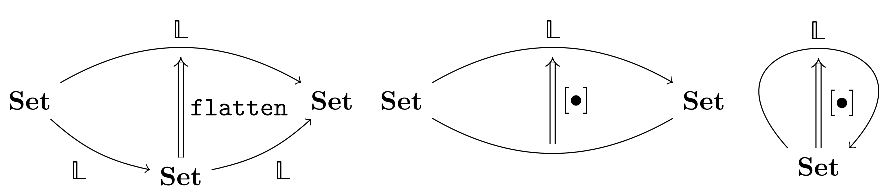
Example. Multicat and **MULTICAT** can be made into 2-categories by using natural transformations as 2-cells. A natural transformation $\alpha : F \Rightarrow G : \mathbf{C} \to \mathbf{D}$ assigns a unary **D**-morphism $[F(C)] \xrightarrow{\alpha_C} [G(C)]$ to each object $C : \mathbf{C}$ in such a way that, for each morphism $m : [C_1, \ldots, C_n] \to C$ of **C**, the **D**-morphism $[\alpha_{C_1}, \ldots, \alpha_{C_n}] ; G(m)$ equals $[F(m)] ; \alpha_C$.

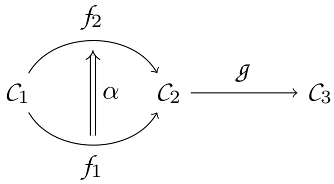
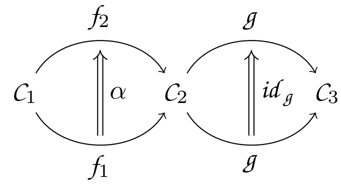
Example. Rel can be made into a 2-category by having a unique 2-cell $X \underset{R_1}{\overset{R_2}{\Uparrow}} Y$ when $\forall x : X, y : Y.\ x\ R_1\ y \Rightarrow x\ R_2\ y$.

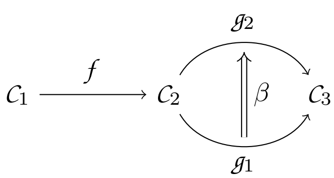
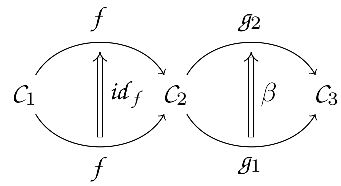
Example. Prost can be made into a 2-category by having a unique 2-cell $\langle X, \leq\rangle \underset{f_1}{\overset{f_2}{\Uparrow}} \langle Y, \leq\rangle$ when $\forall x : X.\ f_1(x) \leq f_2(x)$.

Notation. The identity 2-cell of a morphism f is often denoted by abuse of notation with f. The identity morphism of an object C is often denoted by abuse of notation with C. Note that this means C can also denote the identity 2-cell of its identity morphism.

Notation. Below, the left diagram indicates `flatten` is a natural transformation from $\mathbb{L} ; \mathbb{L}$ to $\mathbb{L}$. The right two diagrams both indicate that $[\bullet]$ (i.e. `singleton`) is a natural transformation from **Set** to $\mathbb{L}$. The unarrowed line in the middle diagram should be viewed as simply stretching a 0-cell. A diagram can similarly have a path of 1-cells as the codomain of a 2-cell.



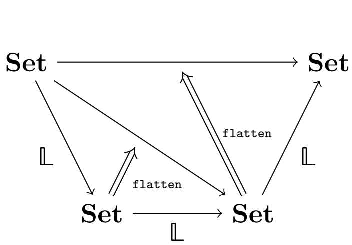
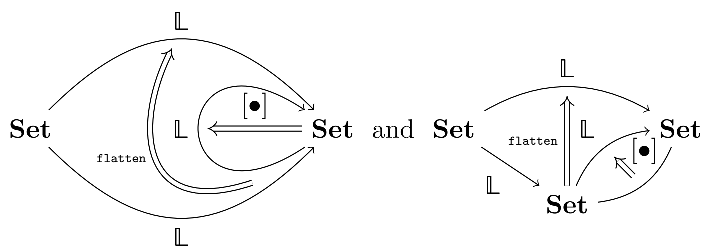
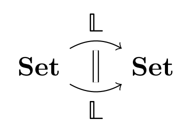
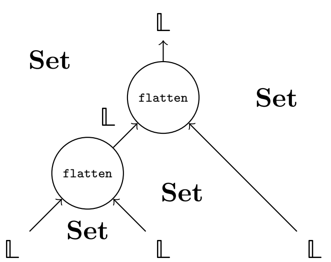
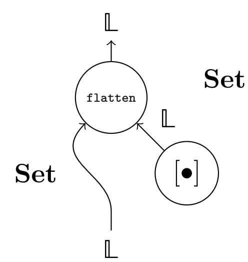
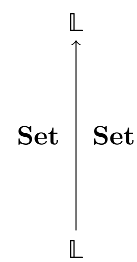
Notation.  denotes the horizontal composition  .

Notation.  denotes the horizontal composition  .

Remark. Due to bifunctoriality of horizontal composition, these notations do not introduce any ambiguity.

Definition (Dual of a Planar Graph)**.** In a planar graph, edges not only connect vertices but also separate faces. Consequently, one can build a planar graph that is called the dual of a planar graph: it has a vertex for each face of the original graph, there is an edge goes from one vertex to another if the original has an edge separating the corresponding faces with the source to the right of the edge, and it has a face for each vertex of the original graph whose border is the edges corresponding to the edges connected to the corresponding vertex.

Notation. One can also denote a 2-cell composition using diagrams dual to what we have presented above:

Cell Diagram			
	can be denoted with	can both be denoted with	with
String Diagram			
Program	$\lambda\vec{\vec{\vec{x}}}.\ \texttt{flatten}(\texttt{map}_{\texttt{flatten}}(\vec{\vec{\vec{x}}}))$	$\lambda\vec{x}.\ \texttt{flatten}([\vec{x}])$	$\lambda\vec{x}.\ \vec{x}$

Remark. Note that all of the above string diagrams could be flipped horizontally to get a new diagram. The following are the programs corresponding to those flipped diagrams.

Flipped Program	$\lambda\vec{\vec{\vec{x}}}.\ \texttt{flatten}(\texttt{flatten}(\vec{\vec{\vec{x}}}))$	$\lambda\vec{x}.\ \texttt{flatten}(\texttt{map}_{[\bullet]}(\vec{x}))$	$\lambda\vec{x}.\ \vec{x}$

Notation. In string diagrams, often the face labels will be elided.

Notation. In string diagrams, often the edge arrows will be elided. Generally, the direction is upward.