(Op)Indexed Categories

Ross Tate

April 25, 2018

Definition (Indexed Category)**.** Given a category $\mathbf{I}$, a (strict) $\mathbf{I}$-indexed category is a functor $\mathbf{C} : \mathbf{I}^{\mathrm{op}} \to \mathbf{Cat}$.

Remark. The intuition is that $\mathbf{I}$ models contexts and assignments between contexts. Given an object I of $\mathbf{I}$, i.e. something modeling a context, $\mathbf{C}(I)$ describes the category under that context. For the case where each $\mathbf{C}(I)$ is thin, i.e. $\mathbf{C}$ is a functor from $\mathbf{I}^{\mathrm{op}}$ to $\mathbf{Prost}$, a common intuition is that the objects of $\mathbf{C}(I)$ model propositions referencing variables in the context I, and a morphism of $\mathbf{C}(I)$ models when one proposition implies another. Given a morphism $a : I \to J$, i.e. something that instantiates the variables in context J with expressions referencing variables in context I, the functor $\mathbf{C}(s)$ conceptually maps propositions ϕ referencing variables in J to their substitution $\phi[a]$, in which every reference to a variable in J is replaced with the corresponding expression specified by a referencing variables in I. Functoriality of $\mathbf{C}$ captures the intuition that $\phi[id]$, i.e. substituting each variable with itself, should simply result in ϕ, and that $\phi[a][a']$, i.e. substituting successively, should result in the same proposition as $\phi[a' \,; a]$, i.e. substituting once with the composed assignment of variables.

Example. Recall that a concrete category $\mathbf{C} \xrightarrow{U} \mathbf{I}$ has a notion of fibres over objects of $\mathbf{I}$. In particular, an object of the fibre over an object I of $\mathbf{I}$ is an object C of $\mathbf{C}$ such that UC equals I, and a morphism $f : C \to C'$ of the fibre over I is a morphism $f : C \to C'$ of $\mathbf{C}$ such that Uf equals id_I. Consequently, there is a function mapping each *object* of $\mathbf{I}$ to a category, namely the fibre over I. However, this does not necessarily extend to a $\mathbf{I}$-indexed category because there may not be any sensible mapping from *morphisms* of $\mathbf{I}$ to functors between the fibres.

Suppose that the concrete category furthermore has the property that every U-structured morphism has a unique initial lifting. Then this additional property can be used to fill in the additional structure needed for a $\mathbf{I}$-indexed category. Given an object D of the fibre over J and a morphism $a : I \to J$ of $\mathbf{I}$, we can view a as a U-structured morphism $a : I \to UD$. By our assumption, this U-structured morphism has a unique initial lifting, say $f : C \to D$, which in particular means that C is an object of the fibre over I. This object C is conceptually the result of substituting D by a, and initiality of f enables this construction to extend to a functor from the fibre over J to the fibre over I, and in such a way that satisfies the requirements of an $\mathbf{I}$-indexed category.

Now if one weakens the property to just existence and not necessarily uniqueness of initial liftings, then construction of these functors between slice categories requires the axiom of choice, and the mapping from $\mathbf{I}^{\mathrm{op}}$ to $\mathbf{Cat}$ is not quite functorial—it is only functorial up to (coherent) isomorphism. As a consequence one more often uses *fibred* category theory rather than indexed category theory. A $\mathbf{I}$-fibred category is a category $\mathbf{C}$ with a (not necessarily faithful) functor to $\mathbf{I}$ that has *cartesian* liftings, where cartesian is the generalization of initial to not-necessarily-faithful functors.

Definition (Opindexed Category)**.** Given a category $\mathbf{I}$, a (strict) $\mathbf{I}$-*op*indexed category is a functor $\mathbf{C} : \mathbf{I} \to \mathbf{Cat}$.

Example. Suppose the objects of $\mathbf{I}$ are conceptually database-table schemas, and the morphisms of $\mathbf{I}$ are conceptually database programs indicating how to generate a row of the target table schema from a row of the source schema. Given a schema I, a table of schema I is conceptually a multiset/bag of rows of schema I. Similarly, a table of schema I is conceptually smaller than another table of schema I if the latter contains (including quantity) all the rows of the former. Or, if one cares about data provenance, a morphism from a table of schema I to another table of schema I is a content-preserving mapping of all the entries of the former to entries of the latter. Thus for every schema I there is a category of tables of schema I. And given a morphism, i.e. database program, from schema I to schema J, there is functor from tables of I to tables of J that maps each table to the table generated by applying the program to each entry of the input table. Clearly this process respects identity and composition of programs, so opindexed category theory is well suited to databases. In fact, the core of the syntax and semantics of, say, C#'s embedded LINQ language follows from this opindexed structure.