Category Theory

Lecture 3

Category-theoretic properties

Any two isomorphic objects in a category should have the same category-theoretic properties – statements that are provable in a formal logic for category theory, whatever that is.

Instead of trying to formalize such a logic, we will just look at examples of category-theoretic properties.

Here is our first one...

Terminal object

An object T of a category $\mathbf{C}$ is terminal if for all $X \in \mathbf{C}$, there is a unique $\mathbf{C}$-morphism from X to T, which we write as $\boxed{\langle\rangle_X : X \to T}$.

So we have $\begin{cases} \forall X \in \mathbf{C},\ \langle\rangle_X \in \mathbf{C}(X, T) \\ \forall X \in \mathbf{C}, \forall f \in \mathbf{C}(X, T),\ f = \langle\rangle_X \end{cases}$

(So in particular, $\mathtt{id}_T = \langle\rangle_T$)

Sometimes we just write $\langle\rangle_X$ as $\langle\rangle$.

Some people write $!_X$ for $\langle\rangle_X$ – there is no commonly accepted notation; [Awodey] avoids using one.

Examples of terminal objects

- In **Set**: any one-element set.
- Any one-element set has a unique pre-order and this makes it terminal in **Preord** (and **Poset**)
- Any one-element set has a unique monoid structure and this makes it terminal in **Mon**.

Examples of terminal objects

- In **Set**: any one-element set.
- Any one-element set has a unique pre-order and this makes it terminal in **Preord** (and **Poset**)
- Any one-element set has a unique monoid structure and this makes it terminal in **Mon**.
- A pre-ordered set $(P, \sqsubseteq)$, regarded as a category $\mathbf{C}_P$, has a terminal object iff it has a greatest element $\top$, that is: $\forall x \in P,\ x \sqsubseteq \top$
- When does a monoid $(M, \cdot, e)$, regarded as a category $\mathbf{C}_M$, have a terminal object?

Terminal object

Theorem. In a category $\mathbf{C}$:

(a) If T is terminal and $T \cong T'$, then T' is terminal.

(b) If T and T' are both terminal, then $T \cong T'$ (and there is only one isomorphism between T and T').

In summary: terminal objects are unique up to unique isomorphism.

Proof...

Terminal object

Theorem. In a category $\mathbf{C}$:

(a) If T is terminal and $T \cong T'$, then T' is terminal.

(b) If T and T' are both terminal, then $T \cong T'$ (and there is only one isomorphism between T and T').

In summary: terminal objects are unique up to unique isomorphism.

Proof...

Notation: from now on, if a category $\mathbf{C}$ has a terminal object we will write that object as $\boxed{1}$

Opposite of a category

Given a category $\mathbf{C}$, its opposite category $\boxed{\mathbf{C}^{\mathrm{op}}}$ is defined by interchanging the operations of $\mathtt{dom}$ and $\mathtt{cod}$ in $\mathbf{C}$:

- $\mathtt{obj}\,\mathbf{C}^{\mathrm{op}} \triangleq \mathtt{obj}\,\mathbf{C}$
- $\mathbf{C}^{\mathrm{op}}(X, Y) \triangleq \mathbf{C}(Y, X)$, for all objects X and Y
- identity morphism on $X \in \mathtt{obj}\,\mathbf{C}^{\mathrm{op}}$ is $\mathtt{id}_X \in \mathbf{C}(X, X) = \mathbf{C}^{\mathrm{op}}(X, X)$
- composition in $\mathbf{C}^{\mathrm{op}}$ of $f \in \mathbf{C}^{\mathrm{op}}(X, Y)$ and $g \in \mathbf{C}^{\mathrm{op}}(Y, Z)$ is given by the composition $f \circ g \in \mathbf{C}(Z, X) = \mathbf{C}^{\mathrm{op}}(X, Z)$ in $\mathbf{C}$ (associativity and unity properties hold for this operation, because they do in $\mathbf{C}$)

The Principle of Duality

Whenever one defines a concept / proves a theorem in terms of commutative diagrams in a category $\mathbf{C}$,

one obtains another concept / theorem, called its dual,

by reversing the direction or morphisms throughout, that is, by replacing $\mathbf{C}$ by its opposite category $\mathbf{C}^{\mathrm{op}}$.

For example…

Initial object

is the dual notion to "terminal object":

An object 0 of a category $\mathbf{C}$ is initial if for all $X \in \mathbf{C}$, there is a unique $\mathbf{C}$-morphism $0 \to X$, which we write as $\boxed{[\,]_X : 0 \to X}$.

So we have $\begin{cases} \forall X \in \mathbf{C},\ \ [\,]_X \in \mathbf{C}(0, X) \\ \forall X \in \mathbf{C}, \forall f \in \mathbf{C}(0, X),\ \ f = [\,]_X \end{cases}$

(So in particular, $\mathtt{id}_0 = [\,]_0$)

By duality, we have that initial objects are unique up to isomorphism and that any object isomorphic to an initial object is itself initial.
(**N.B.** "isomorphism" is a self-dual concept.)

Examples of initial objects

- The empty set is initial in **Set**.
- Any one-element set has a uniquely determined monoid structure and is initial in **Mon**. (why?)

  So initial and terminal objects co-incide in **Mon**

  An object that is both initial and terminal in a category is sometimes called a zero object.
- A pre-ordered set $(P, \sqsubseteq)$, regarded as a category $\mathbf{C}_P$, has an initial object iff it has a least element $\perp$, that is: $\forall x \in P, \perp \sqsubseteq x$

Example: free monoids as initial objects

(relevant to automata and formal languages)

The free monoid on a set Σ is $(\mathtt{List}\,\Sigma, @, \mathtt{nil})$ where

$$
\begin{array}{rcl}
\mathtt{List}\,\Sigma & = & \text{set of finite lists of elements of } \Sigma \\
@ & = & \text{list concatenation} \\
\mathtt{nil} & = & \text{empty list}
\end{array}
$$

Example:
free monoids as initial objects

(relevant to automata and formal languages)

The free monoid on a set Σ is $(\mathtt{List}\,\Sigma, @, \mathtt{nil})$ where

$$\begin{array}{rcl} \mathtt{List}\,\Sigma & = & \text{set of finite lists of elements of } \Sigma \\ @ & = & \text{list concatenation} \\ \mathtt{nil} & = & \text{empty list} \end{array}$$

The function

$$\begin{array}{rcl} \eta_\Sigma : \Sigma & \longrightarrow & \mathtt{List}\,\Sigma \\ a & \longmapsto & [a] = a :: \mathtt{nil} \text{ (one-element list)} \end{array}$$

has the following "universal property"...

Example:
free monoids as initial objects

(relevant to automata and formal languages)

Theorem. For any monoid $(M, \cdot, e)$ and function $\underline{f} : \Sigma \to M$, there is a unique monoid morphism $\overline{f} \in \mathbf{Mon}((\mathtt{List}\,\Sigma, @, \mathtt{nil}), (M, \cdot, e))$ making

$$\begin{tikzcd} \Sigma \arrow[r, "\eta_\Sigma"] \arrow[dr, "f"'] & \mathtt{List}\,\Sigma \arrow[d, "\overline{f}"] \\ & M \end{tikzcd}$$

commute in **Set**.

Proof...

Example:
free monoids as initial objects

(relevant to automata and formal languages)

Theorem. $\forall M \in \mathbf{Mon}, \forall f \in \mathbf{Set}(\Sigma, M), \exists! \overline{f} \in \mathbf{Mon}(\mathtt{List}\,\Sigma, M),\ \overline{f} \circ \eta_\Sigma = f$

The theorem just says that $\eta_\Sigma : \Sigma \to \mathtt{List}\,\Sigma$ is an initial object in the category $\Sigma/\mathbf{Mon}$:

- objects: $((M, \cdot, e), f)$ where $(M, \cdot, e) \in \mathtt{obj}\ \mathbf{Mon}$ and $f \in \mathbf{Set}(\Sigma, M)$
- morphisms in $\Sigma/\mathbf{Mon}(((M_1, \cdot_1, e_1), f_1), ((M_2, \cdot_2, e_2), f_2))$ are $f \in \mathbf{Mon}((M_1, \cdot_1, e_1), (M_2, \cdot_2, e_2))$ such that $f \circ f_1 = f_2$
- identities and composition as in $\mathbf{Mon}$

Example: free monoids as initial objects

(relevant to automata and formal languages)

Theorem. $\forall M \in \mathbf{Mon}, \forall f \in \mathbf{Set}(\Sigma, M), \exists! \overline{f} \in \mathbf{Mon}(\mathtt{List}\,\Sigma, M),\ \overline{f} \circ \eta_\Sigma = f$

The theorem just says that $\eta_\Sigma : \Sigma \to \mathtt{List}\,\Sigma$ is an initial object in the category $\Sigma/\mathbf{Mon}$:

So this "universal property" determines the monoid $\mathtt{List}\,\Sigma$ uniquely up to isomorphism in $\mathbf{Mon}$.

We will see later that $\Sigma \mapsto \mathtt{List}\,\Sigma$ is part of a functor (= morphism of categories) which is left adjoint to the "forgetful functor" $\mathbf{Mon} \to \mathbf{Set}$.