Lecture 3
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 $\langle \rangle_X : X \to T$.

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

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

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 $\mathbf{1}$.
Opposite of a category

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

- $\text{obj} \mathbf{C}^{\text{op}} \triangleq \text{obj} \mathbf{C}$
- $\mathbf{C}^{\text{op}}(X, Y) \triangleq \mathbf{C}(Y, X)$, for all objects $X$ and $Y$
- identity morphism on $X \in \text{obj} \mathbf{C}^{\text{op}}$ is $\text{id}_X \in \mathbf{C}(X, X) = \mathbf{C}^{\text{op}}(X, X)$
- composition in $\mathbf{C}^{\text{op}}$ of $f \in \mathbf{C}^{\text{op}}(X, Y)$ and $g \in \mathbf{C}^{\text{op}}(Y, Z)$ is given by the composition $f \circ g \in \mathbf{C}(Z, X) = \mathbf{C}^{\text{op}}(X, Z)$ 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 $\mathbf{C}^{\text{op}}$. 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 $[]_X : 0 \to X$.

So we have $\forall X \in \mathbf{C}, \; [ ]_X \in \mathbf{C}(0, X)$

$\forall X \in \mathbf{C}, \forall f \in \mathbf{C}(0, X), \; f = [ ]_X$

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

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 \textbf{Set}.
- Any one-element set has a uniquely determined monoid structure and is initial in \textbf{Mon}. (why?)

So initial and terminal objects co-incide in \textbf{Mon}.

An object that is both initial and terminal in a category is sometimes called a \textit{zero object}.

- A pre-ordered set \((P, \sqsubseteq)\), regarded as a category \(C_P\), has an initial object iff it has a \textit{least element} \(\bot\), that is: \(\forall x \in P, \bot \sqsubseteq x\)
Example:
free monoids as initial objects
(relevant to automata and formal languages)

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

\[
\begin{align*}
\text{List}\ \Sigma & \ = \ \text{set of finite lists of elements of } \Sigma \\
@ & \ = \ \text{list concatenation} \\
\text{nil} & \ = \ \text{empty list}
\end{align*}
\]
Example:
free monoids as initial objects
(relevant to automata and formal languages)

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

- $\text{List } \Sigma = \text{set of finite lists of elements of } \Sigma$
- $@ = \text{list concatenation}$
- $\text{nil} = \text{empty list}$

The function

$$\eta_{\Sigma} : \Sigma \rightarrow \text{List } \Sigma$$

$$a \mapsto [a] = a :: \text{nil} \quad \text{(one-element list)}$$

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 $f : \Sigma \to M$, there is a unique monoid morphism $\overline{f} \in \text{Mon}((\text{List} \Sigma, @, \text{nil}), (M, \cdot, e))$ making $\eta_{\Sigma}$ commute in $\text{Set}$. 

Proof...
Example:
free monoids as initial objects
(relevant to automata and formal languages)

Theorem.
\[ \forall M \in \text{Mon}, \forall f \in \text{Set}(\Sigma, M), \exists! \overline{f} \in \text{Mon}(\text{List}\Sigma, M), \overline{f} \circ \eta_{\Sigma} = f \]

The theorem just says that \( \eta_{\Sigma} : \Sigma \rightarrow \text{List}\Sigma \) is an initial object in the category \( \Sigma/\text{Mon} \):

- objects: \( (M, f) \) where \( M \in \text{obj Mon} \) and \( f \in \text{Set}(\Sigma, M) \)
- morphisms in \( \Sigma/\text{Mon}((M_1, f_1), (M_2, f_2)) \) are \( f \in \text{Mon}(M_1, M_2) \) such that \( f \circ f_1 = f_2 \)
- identities and composition as in \( \text{Mon} \)
Example:
free monoids as initial objects
(relevant to automata and formal languages)

Theorem.
\[ \forall M \in \text{Mon}, \forall f \in \text{Set}(\Sigma, M), \exists! \bar{f} \in \text{Mon}(\text{List} \Sigma, M), \bar{f} \circ \eta_\Sigma = f \]

The theorem just says that \( \eta_\Sigma : \Sigma \rightarrow \text{List} \Sigma \) is an initial object in the category \( \Sigma/\text{Mon} \):

So this “universal property” determines the monoid \( \text{List} \Sigma \) uniquely up to isomorphism in \( \text{Mon} \).

We will see later that \( \Sigma \mapsto \text{List} \Sigma \) is part of a functor (\( = \) morphism of categories) which is left adjoint to the “forgetful functor” \( \text{Mon} \rightarrow \text{Set} \).