Εισαγωγή στη Θεωρία Κατηγοριών

Χριστίνα Βασιλακοπούλου

Οικονομικό Πανεπιστήμιο Αθηνών

26 Μαρτίου 2015

Μια κατηγορία *(category)* αποτελείται από:

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B \ldots$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B...$
- μια συλλογή μορφισμών *(morphisms)* $f, g, ...$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B...$
- μια συλλογή μορφισμών *(morphisms)* $f, g, ...$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom} f$ και $\mathrm{cod} f$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B...$
- μια συλλογή μορφισμών *(morphisms)* $f, g, ...$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom}f$ και $\mathrm{cod}f$ - συμβολίζουμε $f : \mathrm{dom}f \to \mathrm{cod}f$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B...$
- μια συλλογή μορφισμών *(morphisms)* $f, g, ...$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom}f$ και $\mathrm{cod}f$ - συμβολίζουμε $f : \mathrm{dom}f \to \mathrm{cod}f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \to A$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B \ldots$
- μια συλλογή μορφισμών *(morphisms)* $f, g, \ldots$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom} f$ και $\mathrm{cod} f$ - συμβολίζουμε $f : \mathrm{dom} f \to \mathrm{cod} f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \to A$
- για κάθε ζεύγος μορφισμών (f, g) με $\mathrm{cod} f = \mathrm{dom} g$, τη σύνθεσή τους *(composition)* $g \circ f : \mathrm{dom} f \to \mathrm{cod} g$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B...$
- μια συλλογή μορφισμών *(morphisms)* $f, g, ...$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom} f$ και $\mathrm{cod} f$ - συμβολίζουμε $f : \mathrm{dom} f \to \mathrm{cod} f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \to A$
- για κάθε ζεύγος μορφισμών (f, g) με $\mathrm{cod} f = \mathrm{dom} g$, τη σύνθεσή τους *(composition)* $g \circ f : \mathrm{dom} f \to \mathrm{cod} g$ που ικανοποιεί
  - προσεταιριστικότητα: $(h \circ g) \circ f = h \circ (g \circ f)$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B \ldots$
- μια συλλογή μορφισμών *(morphisms)* $f, g, \ldots$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom} f$ και $\mathrm{cod} f$ - συμβολίζουμε $f : \mathrm{dom} f \rightarrow \mathrm{cod} f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \rightarrow A$
- για κάθε ζεύγος μορφισμών (f, g) με $\mathrm{cod} f = \mathrm{dom} g$, τη σύνθεσή τους *(composition)* $g \circ f : \mathrm{dom} f \rightarrow \mathrm{cod} g$ που ικανοποιεί
  - προσεταιριστικότητα: $(h \circ g) \circ f = h \circ (g \circ f)$
  - ταυτότητα: $\mathrm{id}_B \circ f = f \circ \mathrm{id}_A = f$, για $f : A \rightarrow B$

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B \ldots$
- μια συλλογή μορφισμών *(morphisms)* $f, g, \ldots$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom} f$ και $\mathrm{cod} f$ - συμβολίζουμε $f : \mathrm{dom} f \to \mathrm{cod} f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \to A$
- για κάθε ζεύγος μορφισμών (f, g) με $\mathrm{cod} f = \mathrm{dom} g$, τη σύνθεσή τους *(composition)* $g \circ f : \mathrm{dom} f \to \mathrm{cod} g$ που ικανοποιεί
  - προσεταιριστικότητα: $(h \circ g) \circ f = h \circ (g \circ f)$
  - ταυτότητα: $\mathrm{id}_B \circ f = f \circ \mathrm{id}_A = f$, για $f : A \to B$

Από διαφορετική σκοπιά, μπορούμε να θεωρήσουμε τα αντικείμενα ως φόρμουλες *(formulas)*

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* A, B...
- μια συλλογή μορφισμών *(morphisms)* $f, g, \ldots$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom} f$ και $\mathrm{cod} f$ - συμβολίζουμε $f : \mathrm{dom} f \to \mathrm{cod} f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \to A$
- για κάθε ζεύγος μορφισμών (f, g) με $\mathrm{cod} f = \mathrm{dom} g$, τη σύνθεσή τους *(composition)* $g \circ f : \mathrm{dom} f \to \mathrm{cod} g$ που ικανοποιεί
  - προσεταιριστικότητα: $(h \circ g) \circ f = h \circ (g \circ f)$
  - ταυτότητα: $\mathrm{id}_B \circ f = f \circ \mathrm{id}_A = f$, για $f : A \to B$

Από διαφορετική σκοπιά, μπορούμε να θεωρήσουμε τα αντικείμενα ως φόρμουλες *(formulas)* και τους μορφισμούς ως αποδείξεις *(proofs)*.

Μια κατηγορία *(category)* αποτελείται από:

- μια συλλογή αντικειμένων *(objects)* $A, B...$
- μια συλλογή μορφισμών *(morphisms)* $f, g, ...$
- για κάθε μορφισμό f, δύο αντικείμενα $\mathrm{dom}f$ και $\mathrm{cod}f$ - συμβολίζουμε $f : \mathrm{dom}f \to \mathrm{cod}f$
- για κάθε αντικείμενο A, ένα μορφισμό $\mathrm{id}_A : A \to A$
- για κάθε ζεύγος μορφισμών (f, g) με $\mathrm{cod}f = \mathrm{dom}g$, τη σύνθεσή τους *(composition)* $g \circ f : \mathrm{dom}f \to \mathrm{cod}g$ που ικανοποιεί
  - προσεταιριστικότητα: $(h \circ g) \circ f = h \circ (g \circ f)$
  - ταυτότητα: $\mathrm{id}_B \circ f = f \circ \mathrm{id}_A = f$, για $f : A \to B$

Από διαφορετική σκοπιά, μπορούμε να θεωρήσουμε τα αντικείμενα ως φόρμουλες *(formulas)* και τους μορφισμούς ως αποδείξεις *(proofs)*. $A \to B$ είναι μια απόδειξη της συνεπαγωγής $A \Rightarrow B$!

Παραδείγματα

Παραδείγματα

1 Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους.

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.
2. Η κατηγορία **Grp** με αντικείμενα όλες τις ομάδες και μορφισμούς ομομορφισμούς ομάδων.

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.
2. Η κατηγορία **Grp** με αντικείμενα όλες τις ομάδες και μορφισμούς ομομορφισμούς ομάδων. Παρομοίως, έχουμε τις κατηγορίες **Rng** των δακτυλίων

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.
2. Η κατηγορία **Grp** με αντικείμενα όλες τις ομάδες και μορφισμούς ομομορφισμούς ομάδων. Παρομοίως, έχουμε τις κατηγορίες **Rng** των δακτυλίων, $\mathbf{Mod}_R$ των R-προτύπων κλπ.

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.
2. Η κατηγορία **Grp** με αντικείμενα όλες τις ομάδες και μορφισμούς ομομορφισμούς ομάδων. Παρομοίως, έχουμε τις κατηγορίες **Rng** των δακτυλίων, $\mathbf{Mod}_R$ των R-προτύπων κλπ.
3. Ένα μονοειδές *(monoid)* $(M, \cdot, e)$ είναι ένα σύνολο M με μια διμελή πράξη $\cdot$ και ένα διακεκριμένο στοιχείο e έτσι ώστε $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ και $x \cdot e = e \cdot x = x$.

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.
2. Η κατηγορία **Grp** με αντικείμενα όλες τις ομάδες και μορφισμούς ομομορφισμούς ομάδων. Παρομοίως, έχουμε τις κατηγορίες **Rng** των δακτυλίων, $\mathbf{Mod}_R$ των R-προτύπων κλπ.
3. Ένα μονοειδές *(monoid)* $(M, \cdot, e)$ είναι ένα σύνολο M με μια διμελή πράξη $\cdot$ και ένα διακεκριμένο στοιχείο e έτσι ώστε $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ και $x \cdot e = e \cdot x = x$. Σχηματίζουν την κατηγορία **Mon** με τις συναρτήσεις που διατηρούν $\cdot, e$.

Παραδείγματα

1. Η κατηγορία **Set** με αντικείμενα όλα τα σύνολα και μορφισμούς (ολικές) συναρτήσεις μεταξύ τους. Ταυτοτικοί μορφισμοί είναι οι ταυτοτικές συναρτήσεις $1_A : A \to A$ και σύνθεση είναι η συνήθης συνολοθεωρητική σύνθεση συναρτήσεων.
2. Η κατηγορία **Grp** με αντικείμενα όλες τις ομάδες και μορφισμούς ομομορφισμούς ομάδων. Παρομοίως, έχουμε τις κατηγορίες **Rng** των δακτυλίων, $\mathbf{Mod}_R$ των R-προτύπων κλπ.
3. Ένα μονοειδές *(monoid)* $(M, \cdot, e)$ είναι ένα σύνολο M με μια διμελή πράξη $\cdot$ και ένα διακεκριμένο στοιχείο e έτσι ώστε $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ και $x \cdot e = e \cdot x = x$. Σχηματίζουν την κατηγορία **Mon** με τις συναρτήσεις που διατηρούν $\cdot, e$.

⋆ Μια κατηγορία με ένα μόνο αντικείμενο ‘είναι’ ένα μονοειδές!

Παραδείγματα

4. Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E)

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών

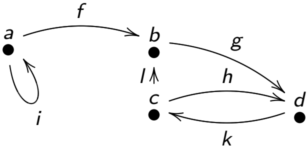
(edges)

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \underset{\text{tail}}{\overset{\text{head}}{\rightrightarrows}} V$.

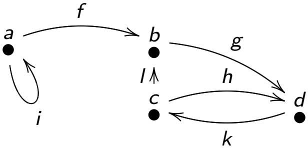
Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \underset{\text{tail}}{\overset{\text{head}}{\rightrightarrows}} V$.



Παραδείγματα

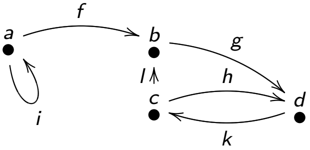
4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \underset{\text{tail}}{\overset{\text{head}}{\rightrightarrows}} V$.



Με τους μορφισμούς γραφών $(t_1, t_2) : (V, E) \to (V', E')$

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \overset{\mathrm{head}}{\underset{\mathrm{tail}}{\rightrightarrows}} V$.



Με τους μορφισμούς γραφών $(t_1, t_2) : (V, E) \to (V', E')$ όπου οι συναρτήσεις t_1, t_2 ικανοποιούν $\mathrm{head}(t_2(f)) = t_1(\mathrm{head}(f))$ και $\mathrm{tail}(t_2(f)) = t_1(\mathrm{tail}(f))$

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \underset{\mathrm{tail}}{\overset{\mathrm{head}}{\rightrightarrows}} V$.

$$\begin{tikzcd}
a \arrow[r, bend left, "f"] \arrow[loop below, "i"] & b \arrow[dr, bend left, "g"] & \\
& c \arrow[u, "l"] \arrow[r, bend left, "h"] & d \arrow[l, bend left, "k"]
\end{tikzcd}$$

Με τους μορφισμούς γραφών $(t_1, t_2) : (V, E) \to (V', E')$ όπου οι συναρτήσεις t_1, t_2 ικανοποιούν $\mathrm{head}(t_2(f)) = t_1(\mathrm{head}(f))$ και $\mathrm{tail}(t_2(f)) = t_1(\mathrm{tail}(f))$ σχηματίζουν την κατηγορία **Grph**.

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \overset{\mathrm{head}}{\underset{\mathrm{tail}}{\rightrightarrows}} V$.

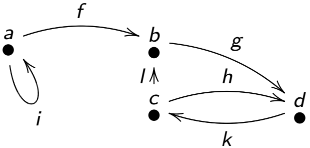
$$\begin{tikzcd} a \arrow[r, bend left, "f"] \arrow[loop below, "i"] & b \arrow[dr, bend left, "g"] & \\ & c \arrow[u, "l"] \arrow[r, bend left, "h"] & d \arrow[l, bend left, "k"] \end{tikzcd}$$

Με τους μορφισμούς γραφών $(t_1, t_2) : (V, E) \to (V', E')$ όπου οι συναρτήσεις t_1, t_2 ικανοποιούν $\mathrm{head}(t_2(f)) = t_1(\mathrm{head}(f))$ και $\mathrm{tail}(t_2(f)) = t_1(\mathrm{tail}(f))$ σχηματίζουν την κατηγορία **Grph**.

Μια κατηγορία είναι ένας κατευθυνόμενος γράφος με δύο επιπλέον λειτουργίες

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \underset{\text{tail}}{\overset{\text{head}}{\rightrightarrows}} V$.

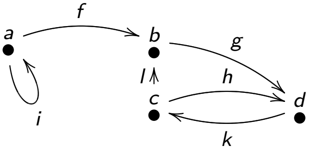


Με τους μορφισμούς γραφών $(t_1, t_2) : (V, E) \rightarrow (V', E')$ όπου οι συναρτήσεις t_1, t_2 ικανοποιούν $\mathrm{head}(t_2(f)) = t_1(\mathrm{head}(f))$ και $\mathrm{tail}(t_2(f)) = t_1(\mathrm{tail}(f))$ σχηματίζουν την κατηγορία **Grph**.

Μια κατηγορία είναι ένας κατευθυνόμενος γράφος με δύο επιπλέον λειτουργίες $V \xrightarrow{\mathrm{id}} E$, $E \times_{\mathrm{comp}} E \xrightarrow{\circ} E$

Παραδείγματα

4 Ένας κατευθυνόμενος γράφος *(directed graph)* είναι ένα ζεύγος (V, E) όπου V είναι το σύνολο των κορυφών ή κόμβων *(nodes)*, E είναι το σύνολο των τόξων ή βελών *(edges)* με δύο συναρτήσεις $E \overset{\text{head}}{\underset{\text{tail}}{\rightrightarrows}} V$.



Με τους μορφισμούς γραφών $(t_1, t_2) : (V, E) \to (V', E')$ όπου οι συναρτήσεις t_1, t_2 ικανοποιούν $\mathrm{head}(t_2(f)) = t_1(\mathrm{head}(f))$ και $\mathrm{tail}(t_2(f)) = t_1(\mathrm{tail}(f))$ σχηματίζουν την κατηγορία **Grph**.

Μια κατηγορία είναι ένας κατευθυνόμενος γράφος με δύο επιπλέον λειτουργίες $V \xrightarrow{\mathrm{id}} E$, $E \times_{\mathrm{comp}} E \xrightarrow{\circ} E$+αξιώματα.

Παραδείγματα

5 Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages)*

Παραδείγματα

5 Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*

Παραδείγματα

5 Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*

6 Η κατηγορία των συναρτήσεων και των ροών δεδομένων *(data flows)*

Παραδείγματα

5. Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*
6. Η κατηγορία των συναρτήσεων και των ροών δεδομένων *(data flows) [data flow diagram]*

Παραδείγματα

5 Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*
6 Η κατηγορία των συναρτήσεων και των ροών δεδομένων *(data flows) [data flow diagram]*
7 Η κατηγορία των τύπων δεδομένων *(data types)* και συναρτήσεων σε δομές δεδομένων *(functions on data structures)*

Παραδείγματα

5. Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*
6. Η κατηγορία των συναρτήσεων και των ροών δεδομένων *(data flows) [data flow diagram]*
7. Η κατηγορία των τύπων δεδομένων *(data types)* και συναρτήσεων σε δομές δεδομένων *(functions on data structures)*
8. Η κατηγορία **Hask** έχει ως αντικείμενα τύπους *Haskell*

Παραδείγματα

5. Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*
6. Η κατηγορία των συναρτήσεων και των ροών δεδομένων *(data flows) [data flow diagram]*
7. Η κατηγορία των τύπων δεδομένων *(data types)* και συναρτήσεων σε δομές δεδομένων *(functions on data structures)*
8. Η κατηγορία **Hask** έχει ως αντικείμενα τύπους *Haskell* και ένας μορφισμός από το A στο B είναι συνάρτηση *Haskell* τύπου $A \to B$.

Παραδείγματα

5 Η κατηγορία των καταστάσεων *(states)* και μηνυμάτων *(messages) [state diagram]*

6 Η κατηγορία των συναρτήσεων και των ροών δεδομένων *(data flows) [data flow diagram]*

7 Η κατηγορία των τύπων δεδομένων *(data types)* και συναρτήσεων σε δομές δεδομένων *(functions on data structures)*

8 Η κατηγορία **Hask** έχει ως αντικείμενα τύπους *Haskell* και ένας μορφισμός από το A στο B είναι συνάρτηση *Haskell* τύπου $A \to B$. Ο ταυτοτικός μορφισμός είναι $\mathrm{id} :: A \to A$ και η σύνθεση είναι $f.g = \backslash x \to f(gx)$.

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$.

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$
- μια απεικόνιση $f \mapsto Ff$ μεταξύ των μορφισμών της $\mathcal{C}$ και της $\mathcal{D}$

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$
- μια απεικόνιση $f \mapsto Ff$ μεταξύ των μορφισμών της $\mathcal{C}$ και της $\mathcal{D}$

τέτοιες ώστε

- διατηρείται το πεδίο και το συν-πεδίο

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$
- μια απεικόνιση $f \mapsto Ff$ μεταξύ των μορφισμών της $\mathcal{C}$ και της $\mathcal{D}$

τέτοιες ώστε

- διατηρείται το πεδίο και το συν-πεδίο, δηλαδή $\mathrm{dom}(Ff) = F\mathrm{dom}f$ και $\mathrm{cod}(Ff) = F\mathrm{cod}f$

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$
- μια απεικόνιση $f \mapsto Ff$ μεταξύ των μορφισμών της $\mathcal{C}$ και της $\mathcal{D}$

τέτοιες ώστε

- διατηρείται το πεδίο και το συν-πεδίο, δηλαδή $\mathrm{dom}(Ff) = F\mathrm{dom}f$ και $\mathrm{cod}(Ff) = F\mathrm{cod}f$
- διατηρείται ο ταυτοτικός μορφισμός και η σύνθεση μορφισμών

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$
- μια απεικόνιση $f \mapsto Ff$ μεταξύ των μορφισμών της $\mathcal{C}$ και της $\mathcal{D}$

τέτοιες ώστε

- διατηρείται το πεδίο και το συν-πεδίο, δηλαδή $\mathrm{dom}(Ff) = F\mathrm{dom}f$ και $\mathrm{cod}(Ff) = F\mathrm{cod}f$
- διατηρείται ο ταυτοτικός μορφισμός και η σύνθεση μορφισμών, δηλαδή $F(\mathrm{id}_A) = \mathrm{id}_{FA}$ και $F(g \circ f) = Fg \circ Ff$.

Έστω δύο κατηγορίες $\mathcal{C}$ και $\mathcal{D}$. Ένας συναρτητής *(functor)* $F : \mathcal{C} \to \mathcal{D}$ αποτελείται από

- μια απεικόνιση $A \mapsto FA$ μεταξύ των αντικειμένων της $\mathcal{C}$ και της $\mathcal{D}$
- μια απεικόνιση $f \mapsto Ff$ μεταξύ των μορφισμών της $\mathcal{C}$ και της $\mathcal{D}$

τέτοιες ώστε

- διατηρείται το πεδίο και το συν-πεδίο, δηλαδή $\mathrm{dom}(Ff) = F\mathrm{dom}f$ και $\mathrm{cod}(Ff) = F\mathrm{cod}f$
- διατηρείται ο ταυτοτικός μορφισμός και η σύνθεση μορφισμών, δηλαδή $F(\mathrm{id}_A) = \mathrm{id}_{FA}$ και $F(g \circ f) = Fg \circ Ff$.

Ουσιαστικά είναι μια συνάρτηση γράφων που διατηρεί την επιπλέον δομή!

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση.

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

⋆ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$

Παραδείγματα

1. Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2. Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

⋆ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$, $L : \mathbf{Cat} \to \mathbf{Grph}$ κλπ.

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

⋆ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$, $L : \mathbf{Cat} \to \mathbf{Grph}$ κλπ.

3 Ο ελεύθερος συναρτητής *(free functor)* $F : \mathbf{Set} \to \mathbf{Grp}$

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$, $L : \mathbf{Cat} \to \mathbf{Grph}$ κλπ.

3 Ο ελεύθερος συναρτητής *(free functor)* $F : \mathbf{Set} \to \mathbf{Grp}$ που στέλνει ένα σύνολο X στην ελεύθερη ομάδα FX

Παραδείγματα

1. Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

⋆ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2. Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$, $L : \mathbf{Cat} \to \mathbf{Grph}$ κλπ.
3. Ο ελεύθερος συναρτητής *(free functor)* $F : \mathbf{Set} \to \mathbf{Grp}$ που στέλνει ένα σύνολο X στην ελεύθερη ομάδα FX, που σχηματίζεται από όλες τις πεπερασμένες ακολουθίες γραμμάτων από το ‘αλφάβητο’ X.

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

$\star$ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$, $L : \mathbf{Cat} \to \mathbf{Grph}$ κλπ.

3 Ο ελεύθερος συναρτητής *(free functor)* $F : \mathbf{Set} \to \mathbf{Grp}$ που στέλνει ένα σύνολο X στην ελεύθερη ομάδα FX, που σχηματίζεται από όλες τις πεπερασμένες ακολουθίες γραμμάτων από το ‘αλφάβητο’ X. Αναλόγως $\mathbf{Set} \to \mathbf{Mon}$

Παραδείγματα

1 Για κάθε κατηγορία $\mathcal{C}$, ο ταυτοτικός συναρτητής $\mathbf{1}_{\mathcal{C}} : \mathcal{C} \to \mathcal{C}$.

⋆ Οι κατηγορίες και οι συναρτητές σχηματίζουν με τη σειρά τους μια κατηγορία **Cat**.

2 Ο επιλήσμων συναρτητής *(forgetful functor)* $U : \mathbf{Grp} \to \mathbf{Set}$ που στέλνει μια ομάδα A στο υποκείμενο σύνολο και έναν ομομορφισμό ομάδων $f : A \to B$ στην αντίστοιχη συνάρτηση. Παρομοίως έχουμε συναρτητές $G : \mathbf{Rng} \to \mathbf{Set}$, $H : \mathbf{Mod}_R \to \mathbf{Set}$, $K : \mathbf{Mon} \to \mathbf{Set}$, $L : \mathbf{Cat} \to \mathbf{Grph}$ κλπ.

3 Ο ελεύθερος συναρτητής *(free functor)* $F : \mathbf{Set} \to \mathbf{Grp}$ που στέλνει ένα σύνολο X στην ελεύθερη ομάδα FX, που σχηματίζεται από όλες τις πεπερασμένες ακολουθίες γραμμάτων από το ‘αλφάβητο’ X. Αναλόγως $\mathbf{Set} \to \mathbf{Mon}$, $\mathbf{Grph} \to \mathbf{Cat}$ *(free category generated by a graph)*.

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του.

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\mathfrak{P} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\begin{tikzcd} \mathfrak{P} : \mathbf{Set} \arrow[r] & \mathbf{Set} \\ S \arrow[r, mapsto, dotted] & \mathfrak{P}S \end{tikzcd}$$

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\mathfrak{P} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

$$\begin{tikzcd}
S \arrow[r, mapsto, dotted] \arrow[d, "f"'] & \mathfrak{P}S \arrow[d, "\mathfrak{P}f"] \\
S' \arrow[r, mapsto, dotted] & \mathfrak{P}S'
\end{tikzcd}$$

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\begin{tikzcd} \mathfrak{P} : \mathbf{Set} \arrow[r] & \mathbf{Set} \\ S \arrow[r, mapsto, dotted] \arrow[d, "f"'] & \mathfrak{P}S \arrow[d, "\mathfrak{P}f"] \\ S' \arrow[r, mapsto, dotted] & \mathfrak{P}S' \end{tikzcd}$$

όπου για $A \subseteq S$, $\mathfrak{P}f(A) = \{f(a) \ / \ a \in A\} \subseteq S'$.

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\mathfrak{P} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

$$\begin{tikzcd} S \arrow[r, mapsto, dotted] \arrow[d, "f"'] & \mathfrak{P}S \arrow[d, "\mathfrak{P}f"] \\ S' \arrow[r, mapsto, dotted] & \mathfrak{P}S' \end{tikzcd}$$

όπου για $A \subseteq S$, $\mathfrak{P}f(A) = \{f(a) \,/\, a \in A\} \subseteq S'$.

5 Για κάθε σύνολο S, σχηματίζουμε το σύνολο $\mathrm{List}(S)$ με πεπερασμένες λίστες στοιχείων του S.

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\begin{tikzcd} \mathfrak{P} : \mathbf{Set} \arrow[r] & \mathbf{Set} \\ S \arrow[r, mapsto, dotted] \arrow[d, "f"'] & \mathfrak{P}S \arrow[d, "\mathfrak{P}f"] \\ S' \arrow[r, mapsto, dotted] & \mathfrak{P}S' \end{tikzcd}$$

όπου για $A \subseteq S$, $\mathfrak{P}f(A) = \{f(a) \ / \ a \in A\} \subseteq S'$.

5 Για κάθε σύνολο S, σχηματίζουμε το σύνολο $\mathrm{List}(S)$ με πεπερασμένες λίστες στοιχείων του S. Έτσι ορίζεται ένας συναρτητής $\mathrm{List} : \mathbf{Set} \to \mathbf{Set}$ στα αντικείμενα

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\mathfrak{P} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

$$\begin{tikzcd} S \arrow[r, mapsto, dotted] \arrow[d, "f"'] & \mathfrak{P}S \arrow[d, "\mathfrak{P}f"] \\ S' \arrow[r, mapsto, dotted] & \mathfrak{P}S' \end{tikzcd}$$

όπου για $A \subseteq S$, $\mathfrak{P}f(A) = \{f(a) \ / \ a \in A\} \subseteq S'$.

5 Για κάθε σύνολο S, σχηματίζουμε το σύνολο $\mathrm{List}(S)$ με πεπερασμένες λίστες στοιχείων του S. Έτσι ορίζεται ένας συναρτητής $\mathrm{List} : \mathbf{Set} \to \mathbf{Set}$ στα αντικείμενα, ενώ μια συνάρτηση $f : S \to S'$ απεικονίζεται στη συνάρτηση $\mathrm{List}(f) : \mathrm{List}(S) \to \mathrm{List}(S')$

Παραδείγματα

4 Για κάθε σύνολο S, γράφουμε $\mathfrak{P}S$ για το δυναμοσύνολό του. Ορίζουμε ένα συναρτητή

$$\mathfrak{P} : \mathbf{Set} \longrightarrow \mathbf{Set}$$

$$\begin{tikzcd} S \arrow[r, mapsto, dotted] \arrow[d, "f"'] & \mathfrak{P}S \arrow[d, "\mathfrak{P}f"] \\ S' \arrow[r, mapsto, dotted] & \mathfrak{P}S' \end{tikzcd}$$

όπου για $A \subseteq S$, $\mathfrak{P}f(A) = \{f(a) \ / \ a \in A\} \subseteq S'$.

5 Για κάθε σύνολο S, σχηματίζουμε το σύνολο $\mathrm{List}(S)$ με πεπερασμένες λίστες στοιχείων του S. Έτσι ορίζεται ένας συναρτητής $\mathrm{List} : \mathbf{Set} \to \mathbf{Set}$ στα αντικείμενα, ενώ μια συνάρτηση $f : S \to S'$ απεικονίζεται στη συνάρτηση $\mathrm{List}(f) : \mathrm{List}(S) \to \mathrm{List}(S')$ που στέλνει την S-λίστα $L = [s_1, s_2, ..., s_n]$ στην S'-λίστα $\mathrm{maplist}(f)(L) = [f(s_1), f(s_2), ..., f(s_n)]$.

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο.

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M.

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

⋆ Πεπερασμένα αυτόματα *(Finite State Automata)*

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

$\star$ Πεπερασμένα αυτόματα *(Finite State Automata)*, όπου S είναι το σύνολο των καταστάσεων

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

$\star$ Πεπερασμένα αυτόματα *(Finite State Automata)*, όπου S είναι το σύνολο των καταστάσεων, M είναι το ελεύθερο μονοειδές επί του αλφαβήτου

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

$\star$ Πεπερασμένα αυτόματα *(Finite State Automata)*, όπου S είναι το σύνολο των καταστάσεων, M είναι το ελεύθερο μονοειδές επί του αλφαβήτου, F δίνει τη συνάρτηση μετάβασης...

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

$\star$ Πεπερασμένα αυτόματα *(Finite State Automata)*, όπου S είναι το σύνολο των καταστάσεων, M είναι το ελεύθερο μονοειδές επί του αλφαβήτου, F δίνει τη συνάρτηση μετάβασης...

7 *"Databases schemas as categories"*:

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

$\star$ Πεπερασμένα αυτόματα *(Finite State Automata)*, όπου S είναι το σύνολο των καταστάσεων, M είναι το ελεύθερο μονοειδές επί του αλφαβήτου, F δίνει τη συνάρτηση μετάβασης...

7 *"Databases schemas as categories"*: κατηγορία $\mathcal{C}$ με αντικείμενα πίνακες *(tables)*, μορφισμούς στήλες

Παραδείγματα

6 Αν M είναι ένα μονοειδές, ένας οποιοσδήποτε συναρτητής $F : M \to \mathbf{Set}$ ονομάζεται και M-σύνολο. Δίνει ένα σύνολο S ως την εικόνα του μοναδικού αντικειμένου στο M, και συναρτήσεις $S \to S$ ως εικόνες των (ενδο)μορφισμών του M. Για $m \in M$, Fm είναι η m-δράση στο S.

$\star$ Πεπερασμένα αυτόματα *(Finite State Automata)*, όπου S είναι το σύνολο των καταστάσεων, M είναι το ελεύθερο μονοειδές επί του αλφαβήτου, F δίνει τη συνάρτηση μετάβασης...

7 *"Databases schemas as categories"*: κατηγορία $\mathcal{C}$ με αντικείμενα πίνακες *(tables)*, μορφισμούς στήλες, και ένας συναρτητής $F : \mathcal{C} \to \mathbf{Set}$ γεμίζει τους πίνακες με δεδομένα.

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων
- *Data Analysis*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων
- *Data Analysis* - κατηγορικό υπόβαθρο μεθόδων *FCA*, *LSA* κ.ά.

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων
- *Data Analysis* - κατηγορικό υπόβαθρο μεθόδων *FCA*, *LSA* κ.ά.
- *Categorical Logic*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων
- *Data Analysis* - κατηγορικό υπόβαθρο μεθόδων *FCA*, *LSA* κ.ά.
- *Categorical Logic* - semantics, formal methods, automated proof proving, artificial intelligence

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων
- *Data Analysis* - κατηγορικό υπόβαθρο μεθόδων *FCA, LSA* κ.ά.
- *Categorical Logic* - semantics, formal methods, automated proof proving, artificial intelligence
- *Network theory*

Βασικές εφαρμογές Θεωρίας Κατηγοριών στην Επιστήμη των Υπολογιστών

- *Concurrency* - κατηγορία συστημάτων μετάβασης, κατηγορία *Petri nets*
- *Functional programming* - μοναδοειδή *(monads)*
- *Complete Partial Orders* - κατηγορία με αντικείμενα *CPO*, μορφισμούς αύξουσες και συνεχείς συναρτήσεις
- *Lambda calculus* - κατασκευή μοντέλων
- *Data Analysis* - κατηγορικό υπόβαθρο μεθόδων *FCA, LSA* κ.ά.
- *Categorical Logic* - semantics, formal methods, automated proof proving, artificial intelligence
- *Network theory* - κατηγορικό πλαίσιο (διαγράμματα), εφαρμογές σε μηχανική, φυσική, βιολογία, οικολογία κ.ά.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

- Δυϊκότητα *(duality)*:

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*:

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*:

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ' ουσιαστικά ίδιες' κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*:

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ' ουσιαστικά ίδιες' κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$. Π.χ. $F : \mathbf{Set} \rightleftarrows \mathbf{Grp} : U$.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ' ουσιαστικά ίδιες' κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$. Π.χ. $F : \mathbf{Set} \rightleftarrows \mathbf{Grp} : U$.
· Μοναδοειδή *(monads)*:

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$. Π.χ. $F : \mathbf{Set} \rightleftarrows \mathbf{Grp} : U$.
· Μοναδοειδή *(monads)*: (T, μ, η) όπου $T : \mathcal{C} \to \mathcal{C}$, $\mu : T \circ T \Rightarrow T$, $\eta : \mathbf{1}_{\mathcal{C}} \Rightarrow T$ και αξιώματα.

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$. Π.χ. $F : \mathbf{Set} \rightleftarrows \mathbf{Grp} : U$.
· Μοναδοειδή *(monads)*: (T, μ, η) όπου $T : \mathcal{C} \to \mathcal{C}$, $\mu : T \circ T \Rightarrow T$, $\eta : \mathbf{1}_{\mathcal{C}} \Rightarrow T$ και αξιώματα. Π.χ. $(\mathrm{List}, \mu, \eta)$

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ' ουσιαστικά ίδιες' κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$. Π.χ. $F : \mathbf{Set} \rightleftarrows \mathbf{Grp} : U$.
· Μοναδοειδή *(monads)*: (T, μ, η) όπου $T : \mathcal{C} \to \mathcal{C}$, $\mu : T \circ T \Rightarrow T$, $\eta : \mathbf{1}_{\mathcal{C}} \Rightarrow T$ και αξιώματα. Π.χ. $(\mathrm{List}, \mu, \eta)$ όπου μ_S δίνει $[[s_1, \dots, s_n], [t_1, \dots, t_m], \dots, [u_1, .., u_k]] \mapsto [s_1, \dots, u_k]$

Περαιτέρω θεμελιώδεις έννοιες και κατασκευές

· Δυϊκότητα *(duality)*: για κάθε κατηγορία $\mathcal{C}$, η δυϊκή της $\mathcal{C}^{\mathrm{op}}$ έχει ίδια αντικείμενα αλλά μορφισμούς αντίθετης κατεύθυνσης.
· Ισοδύναμες κατηγορίες *(equivalence)*: ‘ ουσιαστικά ίδιες’ κατηγορίες μέσω κατάλληλων συναρτητών, κοινά θεωρήματα.
· Καθολικές κατασκευές, όρια, συνόρια *(universality, limits, colimits)*: κατηγορικά γινόμενα και συνγινόμενα, χωρίς αναφορά στην εσωτερική δομή των αντικειμένων.
· Προσαρτημένοι συναρτητές *(adjunction)*: $F : \mathcal{C} \rightleftarrows \mathcal{D} : G$ με μια 1-1 αντιστοιχία μεταξύ μορφισμών $FC \to D$ στην $\mathcal{D}$ και $C \to GD$ στην $\mathcal{C}$. Π.χ. $F : \mathbf{Set} \rightleftarrows \mathbf{Grp} : U$.
· Μοναδοειδή *(monads)*: (T, μ, η) όπου $T : \mathcal{C} \to \mathcal{C}$, $\mu : T \circ T \Rightarrow T$, $\eta : \mathbf{1}_{\mathcal{C}} \Rightarrow T$ και αξιώματα. Π.χ. $(\mathrm{List}, \mu, \eta)$ όπου μ_S δίνει $[[s_1, ..., s_n], [t_1, ..., t_m], ..., [u_1, .., u_k]] \mapsto [s_1, ..., u_k]$ και η_S δίνει $s \mapsto [s]$.

Ιστορικά, την δεκαετία του ′50 προέκυψαν οι βασικές ιδέες μέσα από την Αλγεβρική Τοπολογία

Ιστορικά, την δεκαετία του $'50$ προέκυψαν οι βασικές ιδέες μέσα από την Αλγεβρική Τοπολογία, αργότερα εξέλιξη με εφαρμογές σε Άλγεβρα, Γεωμετρία, Κβαντική Φυσική, Θεμελίωση των Μαθηματικών, Επιστήμη των υπολογιστών...

Ιστορικά, την δεκαετία του $'50$ προέκυψαν οι βασικές ιδέες μέσα από την Αλγεβρική Τοπολογία, αργότερα εξέλιξη με εφαρμογές σε Άλγεβρα, Γεωμετρία, Κβαντική Φυσική, Θεμελίωση των Μαθηματικών, Επιστήμη των υπολογιστών...

ΙΔΕΑ: Η θεωρία Κατηγοριών ως ένα ισχυρό εργαλείο για την εξέταση ιδιοτήτων και χαρακτηριστικών (μαθηματικών) δομών.

Ιστορικά, την δεκαετία του ′50 προέκυψαν οι βασικές ιδέες μέσα από την Αλγεβρική Τοπολογία, αργότερα εξέλιξη με εφαρμογές σε Άλγεβρα, Γεωμετρία, Κβαντική Φυσική, Θεμελίωση των Μαθηματικών, Επιστήμη των υπολογιστών...

ΙΔΕΑ: Η θεωρία Κατηγοριών ως ένα ισχυρό εργαλείο για την εξέταση ιδιοτήτων και χαρακτηριστικών (μαθηματικών) δομών. Τυποποίηση εννοιών μέσω αφαιρετικής διαδικασίας, με στόχο την ενοποίηση και ταξινόμηση θεωριών ‵ σε ένα υψηλότερο επίπεδο′!

Ευχαριστώ πολύ!