A Concrete Introduction to Abstract Inductive Datatypes

Andrei Popescu

Middlesex University
School of Science and Technology
Foundations of Computing Group

See also

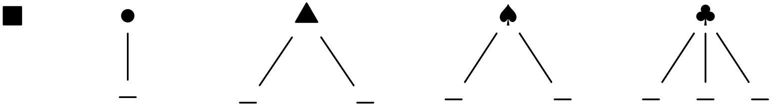
www.andreipopescu.uk/resourcesForStudents/introductionToCodatatypes.pdf

www.andreipopescu.uk/resourcesForStudents/codatatypesInIsabelleHOL.pdf

www.andreipopescu.uk/slides/ESOP2015-slides.pdf

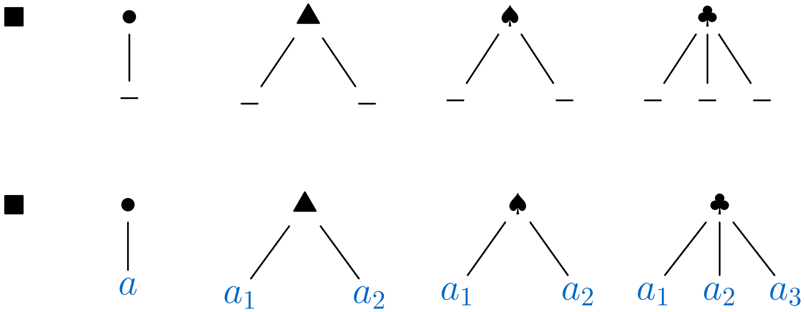
Preliminariers: It’s All About Shape and Content

Shapes



Preliminarieries: It's All About Shape and Content

Shapes



Shapes filled with content from a set $A = \{a_1, a_2, \ldots\}$

Natural Functors on Set

Set = the class of all sets

Natural Functors on Set

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is a natural functor if:

Natural Functors on Set

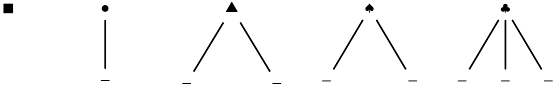
$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is a natural functor if:

It comes with a set of shapes

Natural Functors on Set

F : Set → Set is a natural functor if:

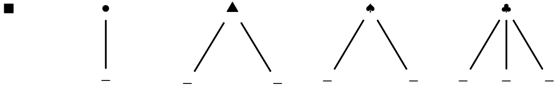
It comes with a set of shapes, say



Natural Functors on Set

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is a natural functor if:

It comes with a set of shapes, say



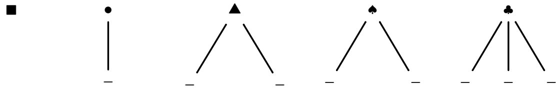
Each element $x \in \mathsf{F}\ A$ consists of:

a choice of a shape

Natural Functors on Set

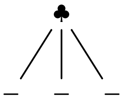
$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is a natural functor if:

It comes with a set of shapes, say



Each element $x \in \mathsf{F}\,A$ consists of:

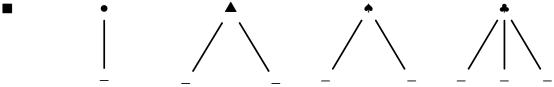
a choice of a shape, say



Natural Functors on Set

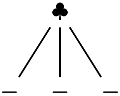
$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is a natural functor if:

It comes with a set of shapes, say



Each element $x \in \mathsf{F}\ A$ consists of:

a choice of a shape, say

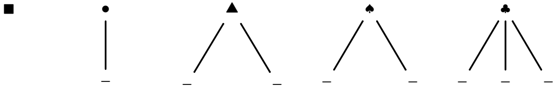


a filling with content from A

Natural Functors on Set

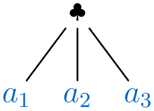
$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$ is a natural functor if:

It comes with a set of shapes, say



Each element $x \in \mathsf{F}\, A$ consists of:

a choice of a shape, say



a filling with content from A, say

Examples of Natural Functors

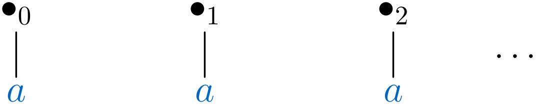
$\mathsf{F}\ A = \mathbb{N} \times A$

Examples of Natural Functors

$\mathsf{F}\ A = \mathbb{N} \times A$

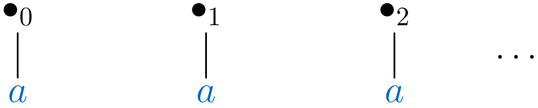
Examples of Natural Functors

$\mathsf{F}\ A = \mathbb{N} \times A$



Examples of Natural Functors

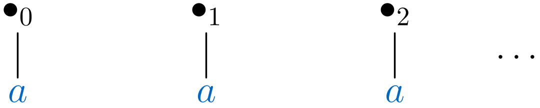
$\mathsf{F}\ A = \mathbb{N} \times A$



$\mathsf{F}\ A = \mathbb{N} + A$

Examples of Natural Functors

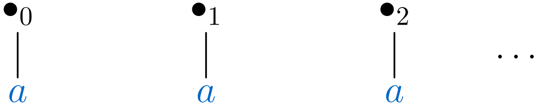
$\mathsf{F}\ A = \mathbb{N} \times A$



$\mathsf{F}\ A = \mathbb{N} + A$

Examples of Natural Functors

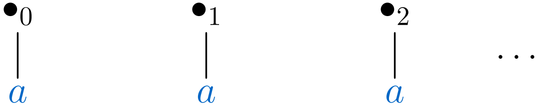
$\mathsf{F}\ A = \mathbb{N} \times A$



$\mathsf{F}\ A = \mathbb{N} + A$

Examples of Natural Functors

$\mathsf{F}\ A = \mathbb{N} \times A$

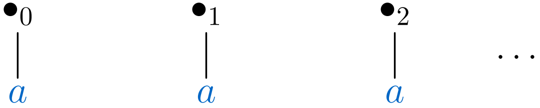


$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F}\ A = \mathsf{List}\ A$

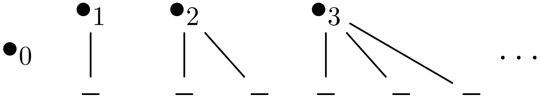
Examples of Natural Functors

$\mathsf{F}\ A = \mathbb{N} \times A$



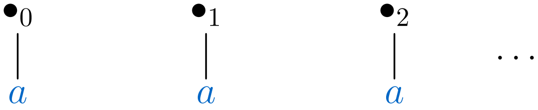
$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F}\ A = \mathsf{List}\ A$

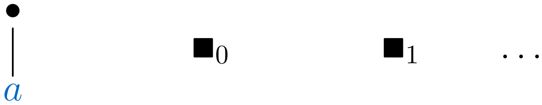


Examples of Natural Functors

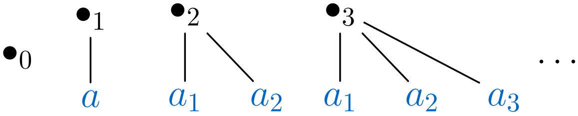
$\mathsf{F}\ A = \mathbb{N} \times A$



$\mathsf{F}\ A = \mathbb{N} + A$

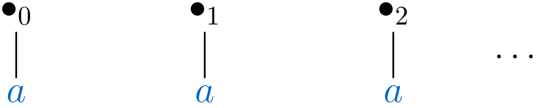


$\mathsf{F}\ A = \mathsf{List}\ A$

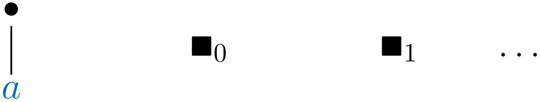


Examples of Natural Functors

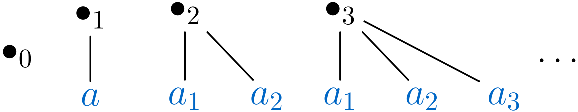
$\mathsf{F}\ A = \mathbb{N} \times A$



$\mathsf{F}\ A = \mathbb{N} + A$



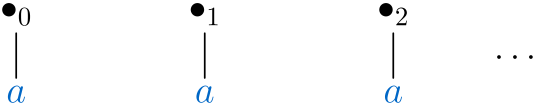
$\mathsf{F}\ A = \mathsf{List}\ A$



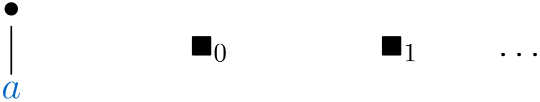
$\mathsf{F}\ A = \mathsf{Stream}\ A$?

Examples of Natural Functors

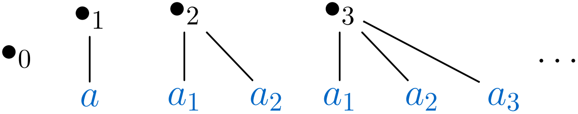
$\mathsf{F}\;A = \mathbb{N} \times A$



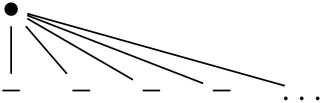
$\mathsf{F}\;A = \mathbb{N} + A$



$\mathsf{F}\;A = \mathsf{List}\;A$

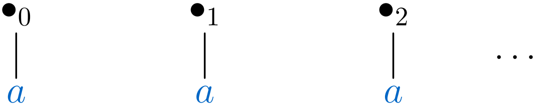


$\mathsf{F}\;A = \mathsf{Stream}\;A$



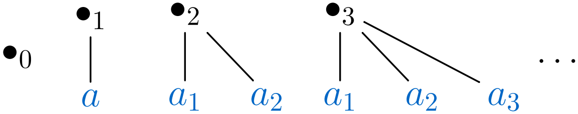
Examples of Natural Functors

$\mathsf{F}\ A = \mathbb{N} \times A$

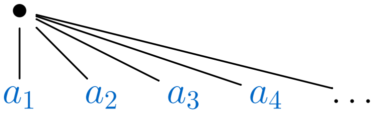


$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F}\ A = \mathsf{List}\ A$



$\mathsf{F}\ A = \mathsf{Stream}\ A$

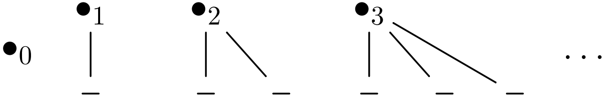


Examples of Natural Functors

F A = Lazy_List A ?

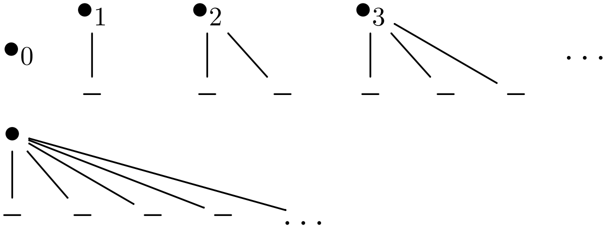
Examples of Natural Functors

F A = Lazy_List A = List A



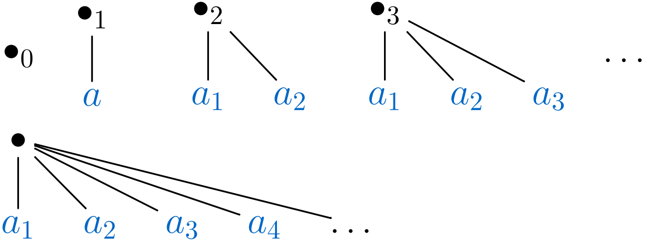
Examples of Natural Functors

$\mathsf{F}\ A = \mathsf{Lazy\_List}\ A\ = \mathsf{List}\ A \cup \mathsf{Stream}\ A$



Examples of Natural Functors

F A = Lazy_List A = List A ∪ Stream A

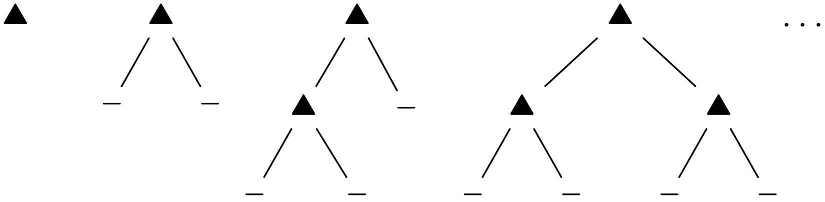


Examples of Natural Functors

$\mathsf{F}\ A = \mathsf{BTree}\ A$ (Full Binary Trees with leaves in A)

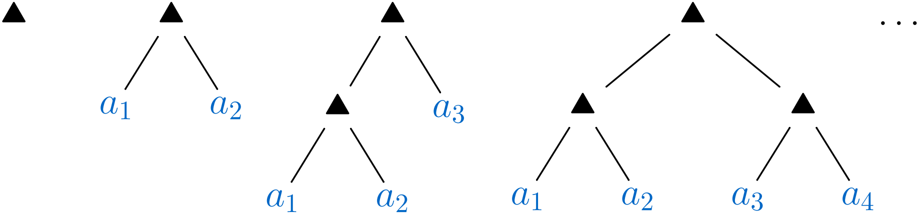
Examples of Natural Functors

$\mathsf{F}\ A = \mathsf{BTree}\ A$ (Full Binary Trees with leaves in A)



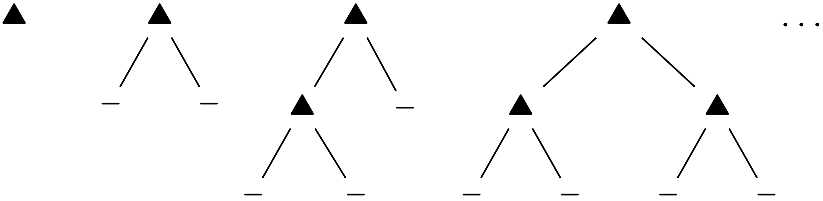
Examples of Natural Functors

F A = BTree A (Full Binary Trees with leaves in A)



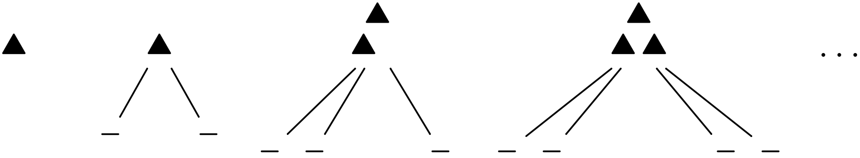
Examples of Natural Functors

F A = BTree A (Full Binary Trees with leaves in A)



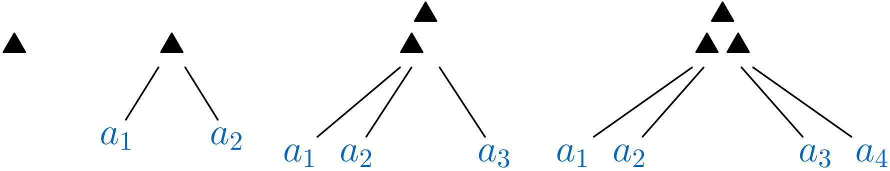
Examples of Natural Functors

$\mathsf{F}\ A = \mathsf{BTree}\ A$ (Full Binary Trees with leaves in A)



Examples of Natural Functors

F A = BTree A (Full Binary Trees with leaves in A)



Functorial Action (Mapper)

$$\begin{tikzcd}
A \arrow[d, "f"] \\
B
\end{tikzcd}$$

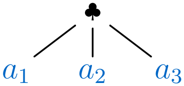
Functorial Action (Mapper)

$$\begin{tikzcd}
A \arrow[d, "f"] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,f"] \\
B & \mathsf{F}\,B
\end{tikzcd}$$

Functorial Action (Mapper)

$$\begin{tikzcd} A \arrow[d, "f"] \\ B \end{tikzcd}$$

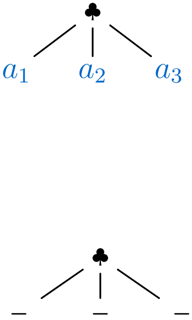
$$\begin{tikzcd} \mathsf{F}\, A \arrow[d, "\mathsf{F}\, f"] \\ \mathsf{F}\, B \end{tikzcd}$$



Functorial Action (Mapper)

$$\begin{tikzcd} A \arrow[d, "f"] \\ B \end{tikzcd}$$

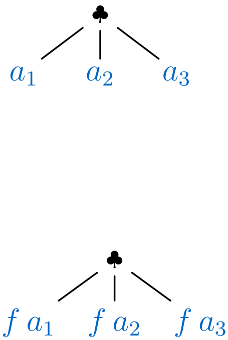
$$\begin{tikzcd} \mathsf{F}\ A \arrow[d, "\mathsf{F}\ f"] \\ \mathsf{F}\ B \end{tikzcd}$$



Keep the same shape

Functorial Action (Mapper)

$$\begin{tikzcd}
A \arrow[d, "f"] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,f"] \\
B & \mathsf{F}\,B
\end{tikzcd}$$

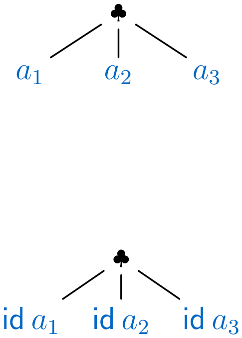


Keep the same shape

Apply f to the content

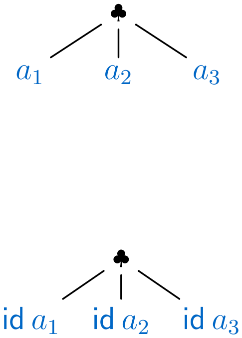
Commutation with the Identity Function

$$\begin{tikzcd}
A \arrow[d, "\mathsf{id}"] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,\mathsf{id}"] \\
A & \mathsf{F}\,A
\end{tikzcd}$$



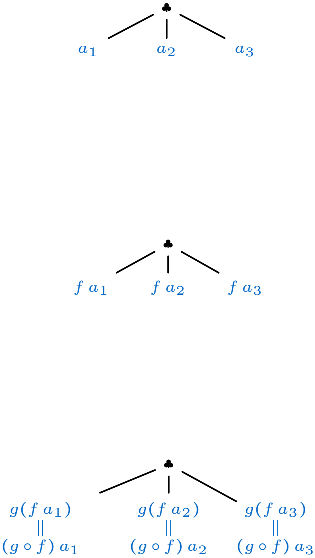
Commutation with the Identity Function

$$\begin{tikzcd}
A \arrow[d, "\mathsf{id}"] & \mathsf{F}\,A \arrow[d, "\mathsf{F}\,\mathsf{id} = \mathsf{id}"] \\
A & \mathsf{F}\,A
\end{tikzcd}$$



Commutation with Function Composition

$$\begin{tikzcd}
A \arrow[d, "f"] & & \mathsf{F}\, A \arrow[d, "\mathsf{F}\, f"] \arrow[dd, bend right, "\mathsf{F}\,(g \circ f)"'] \\
B \arrow[d, "g"] & & \mathsf{F}\, B \arrow[d, "\mathsf{F}\, g"] \\
C & & \mathsf{F}\, C
\end{tikzcd}$$



Bottom Line

$\mathsf{F} : \mathsf{Set} \rightarrow \mathsf{Set}$

For all $A \xrightarrow{f} B$, we have $\mathsf{F}\,A \xrightarrow{\mathsf{F}\,f} \mathsf{F}\,B$ such that:

$$\mathsf{F}\ \mathsf{id}_A = \mathsf{id}_{\mathsf{F}\,A}$$
$$\mathsf{F}\,(g \circ f) = \mathsf{F}\,g \circ \mathsf{F}\,f$$

Bottom Line

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

For all $A \xrightarrow{f} B$, we have $\mathsf{F}\,A \xrightarrow{\mathsf{F}\,f} \mathsf{F}\,B$ such that:

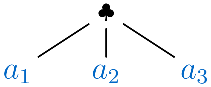
$$\begin{aligned}\mathsf{F}\,\mathsf{id}_A &= \mathsf{id}_{\mathsf{F}\,A}\\ \mathsf{F}\,(g \circ f) &= \mathsf{F}\,g \circ \mathsf{F}\,f\end{aligned}$$

Functoriality

Atoms

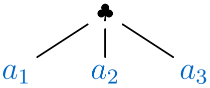
$$\begin{tikzcd}\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] & \mathcal{P}\ A\end{tikzcd}$$

Atoms



$$\begin{tikzcd} \mathsf{F}\,A \arrow[r, "\mathsf{Fatoms}_A"] & \mathcal{P}\,A \end{tikzcd}$$

Atoms



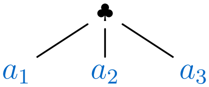
$$\{a_1, a_2, a_3\}$$

$$\mathsf{F}\,A \xrightarrow{\mathsf{Fatoms}_A} \mathcal{P}\,A$$

Atoms

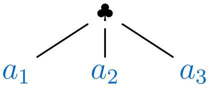
$$\begin{tikzcd}
\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] \arrow[d, "\mathsf{F}\ f"'] & \mathcal{P}\ A \arrow[d, "\mathsf{image}\ f"] \\
\mathsf{F}\ B \arrow[r, "\mathsf{Fatoms}_B"] & \mathcal{P}\ B
\end{tikzcd}$$

Atoms



$$\begin{tikzcd}
\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] \arrow[d, "\mathsf{F}\ f"'] & \mathcal{P}\ A \arrow[d, "\mathsf{image}\ f"] \\
\mathsf{F}\ B \arrow[r, "\mathsf{Fatoms}_B"] & \mathcal{P}\ B
\end{tikzcd}$$

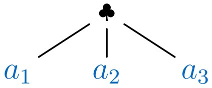
Atoms



$\{a_1, a_2, a_3\}$

$$\begin{tikzcd}
\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] \arrow[d, "\mathsf{F}\ f"'] & \mathcal{P}\ A \arrow[d, "\mathsf{image}\ f"] \\
\mathsf{F}\ B \arrow[r, "\mathsf{Fatoms}_B"] & \mathcal{P}\ B
\end{tikzcd}$$

Atoms

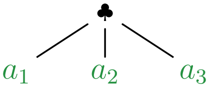


$\{a_1, a_2, a_3\}$

$$\begin{tikzcd}
\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] \arrow[d, "\mathsf{F}\ f"'] & \mathcal{P}\ A \arrow[d, "\mathsf{image}\ f"] \\
\mathsf{F}\ B \arrow[r, "\mathsf{Fatoms}_B"] & \mathcal{P}\ B
\end{tikzcd}$$

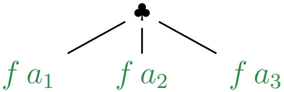
$\{f\ a_1, f\ a_2, f\ a_3\}$

Atoms



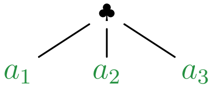
$\{a_1, a_2, a_3\}$

$$\begin{tikzcd}
\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] \arrow[d, "\mathsf{F}\ f"'] & \mathcal{P}\ A \arrow[d, "\mathsf{image}\ f"] \\
\mathsf{F}\ B \arrow[r, "\mathsf{Fatoms}_B"] & \mathcal{P}\ B
\end{tikzcd}$$



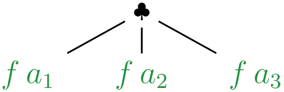
$\{f\ a_1, f\ a_2, f\ a_3\}$

Atoms



$\{a_1, a_2, a_3\}$

$$\begin{tikzcd}
\mathsf{F}\ A \arrow[r, "\mathsf{Fatoms}_A"] \arrow[d, "\mathsf{F}\ f"'] & \mathcal{P}\ A \arrow[d, "\mathsf{image}\ f"] \\
\mathsf{F}\ B \arrow[r, "\mathsf{Fatoms}_B"] & \mathcal{P}\ B
\end{tikzcd}$$



$\{f\ a_1, f\ a_2, f\ a_3\}$

Bottom Line

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

For all $A \xrightarrow{f} B$, we have $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ such that:

$\mathsf{F}\ \mathsf{id}_A = \mathsf{id}_{\mathsf{F}A}$
$\mathsf{F}\ (g \circ f) = \mathsf{F}\ g \circ \mathsf{F}\ f$

Functoriality

Bottom Line

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

For all $A \xrightarrow{f} B$, we have $\mathsf{F}\,A \xrightarrow{\mathsf{F}\,f} \mathsf{F}\,B$ such that:

$$\begin{aligned} \mathsf{F}\,\mathsf{id}_A &= \mathsf{id}_{\mathsf{F}A} \\ \mathsf{F}\,(g \circ f) &= \mathsf{F}\,g \circ \mathsf{F}\,f \end{aligned}$$

Functoriality

For all A, we have $\mathsf{F}\,A \xrightarrow{\mathsf{Fatoms}_A} \mathcal{P}\,A$ such that, for all $A \xrightarrow{f} B$:

$$\mathsf{image}\,f \circ \mathsf{Fatoms}_A = \mathsf{Fatoms}_B \circ \mathsf{image}\,f$$

Bottom Line

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

For all $A \overset{f}{\to} B$, we have $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ such that:

$$\begin{aligned} &\mathsf{F}\ \mathsf{id}_A = \mathsf{id}_{\mathsf{F}A} \\ &\mathsf{F}\ (g \circ f) = \mathsf{F}\ g \circ \mathsf{F}\ f \end{aligned}$$

Functoriality

For all A, we have $\mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}_A} \mathcal{P}\ A$ such that, for all $A \overset{f}{\to} B$:

$$\mathsf{image}\ f \circ \mathsf{Fatoms}_A = \mathsf{Fatoms}_B \circ \mathsf{image}\ f$$

Naturality

Bottom Line: Natural Functors

$\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

For all $A \overset{f}{\to} B$, we have $\mathsf{F}\,A \xrightarrow{\mathsf{F}\,f} \mathsf{F}\,B$ such that:

$\mathsf{F}\,\mathsf{id}_A = \mathsf{id}_{\mathsf{F}\,A}$
$\mathsf{F}\,(g \circ f) = \mathsf{F}\,g \circ \mathsf{F}\,f$ Functoriality

For all A, we have $\mathsf{F}\,A \xrightarrow{\mathsf{Fatoms}_A} \mathcal{P}\,A$ such that, for all $A \overset{f}{\to} B$:

$\mathsf{image}\,f \circ \mathsf{Fatoms}_A = \mathsf{Fatoms}_B \circ \mathsf{image}\,f$ Naturality

Examples of Natural Functors

$A \xrightarrow{f} B$

Examples of Natural Functors

$A \xrightarrow{f} B$ $\mathsf{F}\,A \xrightarrow{\mathsf{F}\,f} \mathsf{F}\,B$

Examples of Natural Functors

$A \xrightarrow{f} B$ $\quad$ $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ $\quad$ $\mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

Examples of Natural Functors

$A \xrightarrow{f} B$ $\quad$ $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ $\quad$ $\mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

Examples of Natural Functors

$A \xrightarrow{f} B$ $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ $\mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$

Examples of Natural Functors

$$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$$

$$\mathsf{F}\ A = \mathbb{N} \times A$$

$$\mathsf{F} f\ (n, a) = (n, f\ a)$$
$$\mathsf{Fatoms}\ (n, a) = \{a\}$$

Examples of Natural Functors

$A \xrightarrow{f} B$ $\quad$ $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ $\quad$ $\mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$
$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

Examples of Natural Functors

$$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$

$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$

Examples of Natural Functors

$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$

$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$

$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing \qquad \mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$

Examples of Natural Functors

$$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$

$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$

$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing \qquad \mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$

$\mathsf{F}\ A = \mathsf{List}\ A$

Examples of Natural Functors

$$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$$

$\mathsf{F}\ A = \mathbb{N} \times A$

$$\mathsf{F}\, f\ (n, a) = (n, f\ a)$$
$$\mathsf{Fatoms}\ (n, a) = \{a\}$$

$\mathsf{F}\ A = \mathbb{N} + A$

$$\mathsf{F}\, f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F}\, f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$$
$$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing \qquad \mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$$

$\mathsf{F}\ A = \mathsf{List}\ A$

$$\mathsf{F}\, f\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = f\ a_1 \cdot f\ a_2 \ldots \cdot f\ a_n$$

Examples of Natural Functors

$$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$
$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$
$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing \qquad \mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$

$\mathsf{F}\ A = \mathsf{List}\ A$

$\mathsf{F} f\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = f\ a_1 \cdot f\ a_2 \ldots \cdot f\ a_n$
$\mathsf{Fatoms}\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = \{a_1, a_2, \ldots, a_n\}$

Examples of Natural Functors

$A \xrightarrow{f} B$ $\qquad$ $\mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B$ $\qquad$ $\mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$
$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n$ $\qquad$ $\mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$
$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing$ $\qquad$ $\mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$

$\mathsf{F}\ A = \mathsf{List}\ A$

$\mathsf{F} f\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = f\ a_1 \cdot f\ a_2 \ldots \cdot f\ a_n$
$\mathsf{Fatoms}\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = \{a_1, a_2, \ldots, a_n\}$

$\mathsf{F}\ A = \mathsf{Stream}\ A$

Examples of Natural Functors

$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$
$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$
$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing \qquad \mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$

$\mathsf{F}\ A = \mathsf{List}\ A$

$\mathsf{F} f\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = f\ a_1 \cdot f\ a_2 \ldots \cdot f\ a_n$
$\mathsf{Fatoms}\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = \{a_1, a_2, \ldots, a_n\}$

$\mathsf{F}\ A = \mathsf{Stream}\ A$

$\mathsf{F} f\ ((a_i)_{i \in \mathbb{N}}) = (f\ a_i)_{i \in \mathbb{N}}$

Examples of Natural Functors

$A \xrightarrow{f} B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{F}\ f} \mathsf{F}\ B \qquad \mathsf{F}\ A \xrightarrow{\mathsf{Fatoms}} \mathcal{P}\ A$

$\mathsf{F}\ A = \mathbb{N} \times A$

$\mathsf{F} f\ (n, a) = (n, f\ a)$
$\mathsf{Fatoms}\ (n, a) = \{a\}$

$\mathsf{F}\ A = \mathbb{N} + A$

$\mathsf{F} f\ (\mathsf{Left}\ n) = \mathsf{Left}\ n \qquad \mathsf{F} f\ (\mathsf{Right}\ a) = \mathsf{Right}\ (f\ a)$
$\mathsf{Fatoms}\ (\mathsf{Left}\ n) = \varnothing \qquad \mathsf{Fatoms}\ (\mathsf{Right}\ a) = \{a\}$

$\mathsf{F}\ A = \mathsf{List}\ A$

$\mathsf{F} f\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = f\ a_1 \cdot f\ a_2 \ldots \cdot f\ a_n$
$\mathsf{Fatoms}\ (a_1 \cdot a_2 \cdot \ldots \cdot a_n) = \{a_1, a_2, \ldots, a_n\}$

$\mathsf{F}\ A = \mathsf{Stream}\ A$

$\mathsf{F} f\ ((a_i)_{i \in \mathbb{N}}) = (f\ a_i)_{i \in \mathbb{N}}$
$\mathsf{Fatoms}\ ((a_i)_{i \in \mathbb{N}}) = \{a_i \mid i \in \mathbb{N}\}$

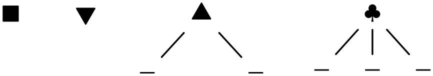
Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

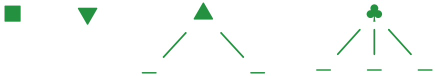
The shapes of F:



Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

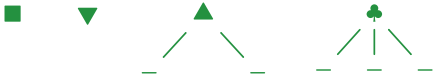
Copies of the shapes of F:



Iterating Shape Composition

Natural functor $F : \mathrm{Set} \to \mathrm{Set}$

Copies of the shapes of F:

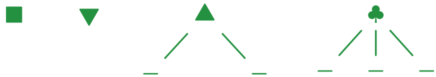


Put them together by plugging in shape for content slot

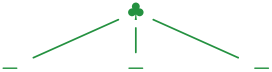
Iterating Shape Composition

Natural functor $F : \mathrm{Set} \to \mathrm{Set}$

Copies of the shapes of F:



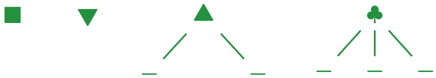
Put them together by plugging in shape for content slot



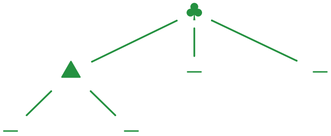
Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



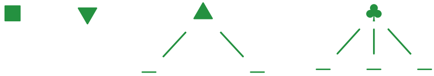
Put them together by plugging in shape for content slot



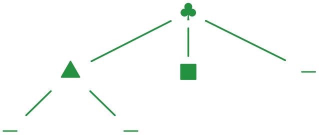
Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



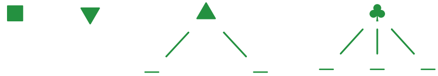
Put them together by plugging in shape for content slot



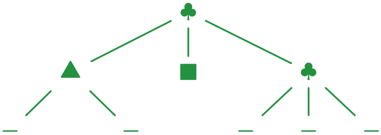
Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



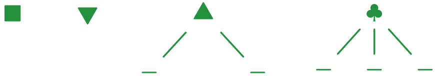
Put them together by plugging in shape for content slot



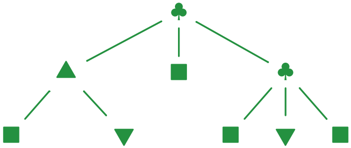
Iterating Shape Composition

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



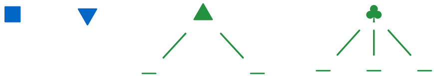
Put them together by plugging in shape for content slot until there are no lingering slots left!



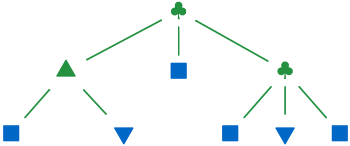
Iterating Shape Composition

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



Put them together by plugging in shape for content slot until there are no lingering slots left!

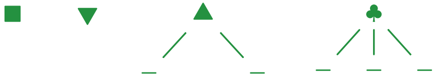


The leaves are always empty-content shapes

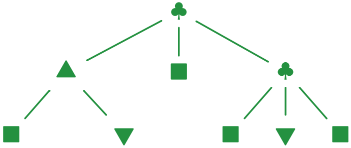
Iterating Shape Composition

Natural functor $F : \mathrm{Set} \to \mathrm{Set}$

Copies of the shapes of F:

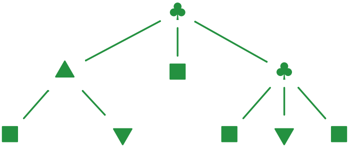


Put them together by plugging in shape for content slot until there are no lingering slots left!



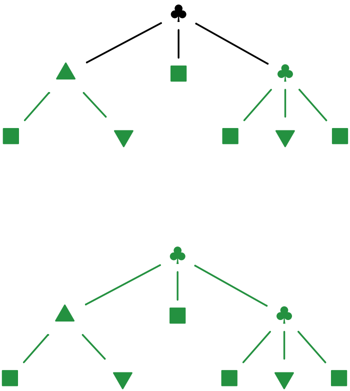
Define I_F = the set of all such finitary couplings

Properties of I_F: Bijectivity



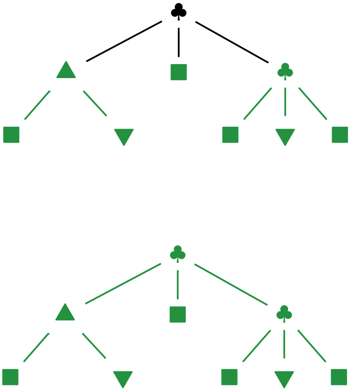
$$\begin{tikzcd} F\, I_F \\ I_F \arrow[u, "\mathsf{dtor}"] \end{tikzcd}$$

Properties of I_F: Bijectivity



$$\begin{tikzcd} F\, I_F \\ I_F \arrow[u, "\mathsf{dtor}"] \end{tikzcd}$$

Properties of I_F: Bijectivity



$$\begin{tikzcd} F\ I_F \arrow[d, shift left, "\text{ctor}"] \\ I_F \arrow[u, shift left, "\text{dtor}"] \end{tikzcd}$$

ctor and dtor are mutually inverse bijections

Properties of I_F: Iteration

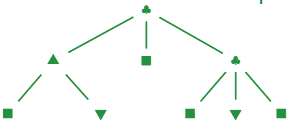
$$\begin{tikzcd}
 & \mathsf{F}\,A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} & A
\end{tikzcd}$$

Properties of I_F: Iteration

$$\begin{tikzcd}
 & \mathsf{F}\,A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} \arrow[r, dotted, "f"] & A
\end{tikzcd}$$

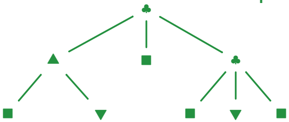
Properties of $\mathsf{I_F}$: Iteration

$$\begin{tikzcd}
 & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I_F} \arrow[r, dotted, "f"] & A
\end{tikzcd}$$

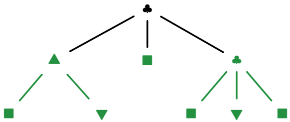


Properties of I_{F}: Iteration

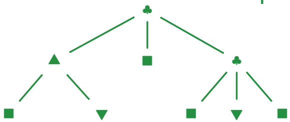
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_{\mathsf{F}} & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_{\mathsf{F}} \arrow[u, "\mathsf{dtor}"'] \arrow[r, dotted, "f"] & A
\end{tikzcd}$$



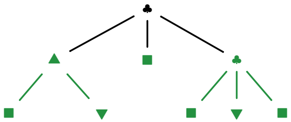
Properties of I_F: Iteration



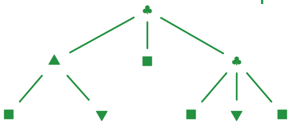
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"'] \arrow[r, dotted, "f"] & A
\end{tikzcd}$$



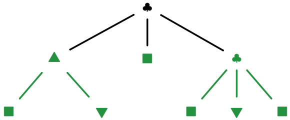
Properties of $\mathsf{I_F}$: Iteration



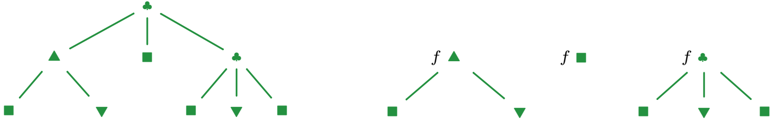
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, dotted, "\mathsf{F}\ f"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I_F} \arrow[u, "\mathsf{dtor}"'] \arrow[r, dotted, "f"] & A
\end{tikzcd}$$



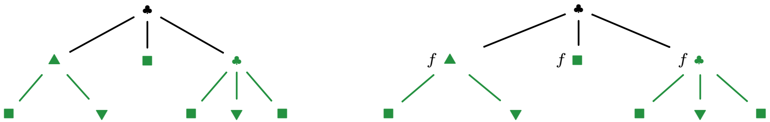
Properties of $\mathsf{I_F}$: Iteration



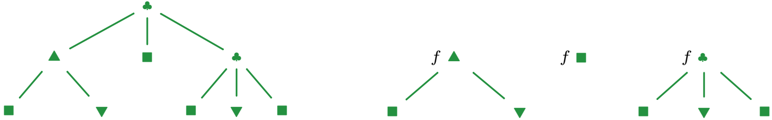
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, dotted, "\mathsf{F}\ f"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I_F} \arrow[u, "\mathsf{dtor}"'] \arrow[r, dotted, "f"] & A
\end{tikzcd}$$



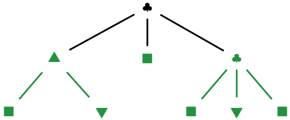
Properties of I_F: Iteration

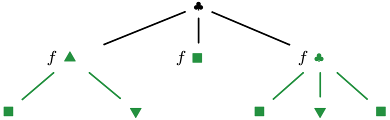


$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, dotted, "\mathsf{F}\ f"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"'] \arrow[r, dotted, "f"] & A
\end{tikzcd}$$

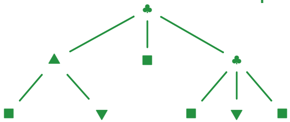


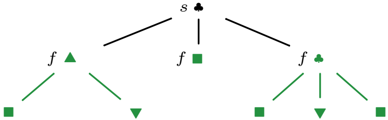
Properties of I_F: Iteration





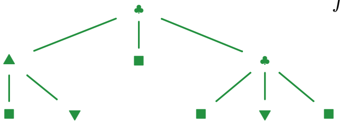
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, dotted, "\mathsf{F}\ f"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"'] \arrow[r, dotted, "f"] & A
\end{tikzcd}$$





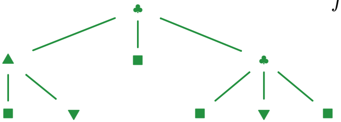
Properties of I_{F}: Iteration

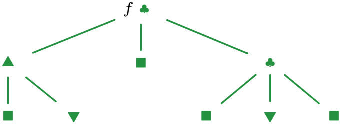
$$\begin{tikzcd}
& \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_{\mathsf{F}} \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$



Properties of I_{F}: Iteration

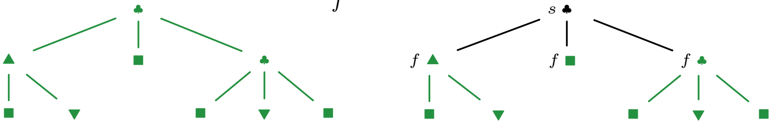
$$\begin{tikzcd}
 & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_{\mathsf{F}} \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$





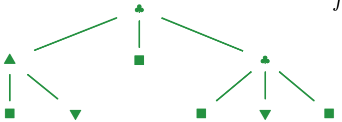
Properties of I_{F}: Iteration

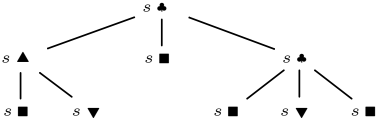
$$\begin{tikzcd}
 & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_{\mathsf{F}} \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$



Properties of I_{F}: Iteration

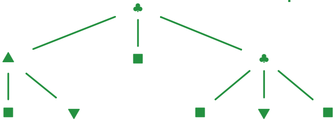
$$\begin{tikzcd} & \mathsf{F}\ A \arrow[d, "s"'] \\ \mathsf{I}_{\mathsf{F}} \arrow[r, dotted, "f"'] & A \end{tikzcd}$$

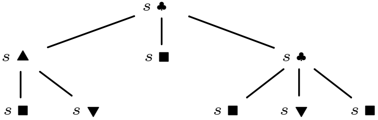




Properties of I_F: Iteration

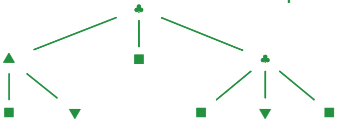
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, dotted, "\mathsf{F}\ f"] & \mathsf{F}\ A \arrow[d, "s"] \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"] \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$

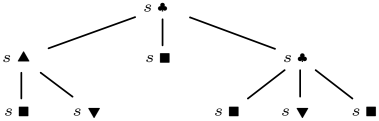




Properties of I_F: Iteration

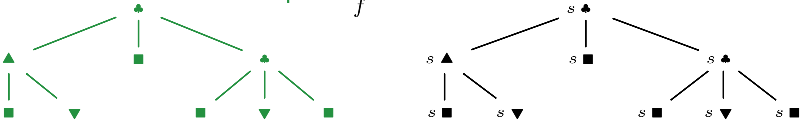
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, dotted, "\mathsf{F}\ f"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$





Properties of I_{F}: Iteration

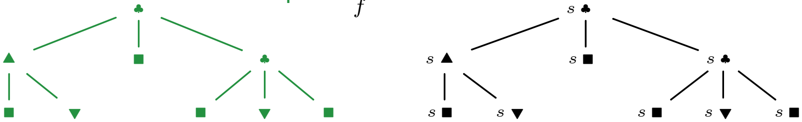
$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_{\mathsf{F}} \arrow[r, dotted, "\mathsf{F}\ f"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"] \\
\mathsf{I}_{\mathsf{F}} \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$



I_{F} is the initial F-algebra

Properties of I_F: Iteration

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, dotted, "\mathsf{F}\ f"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I}_\mathsf{F} \arrow[r, dotted, "f"'] & A
\end{tikzcd}$$



I_F is the initial F-algebra

$f = \mathsf{iter}_s$

Properties of I_F: Induction

I_F

Properties of I_F: Induction

I_F

φ unary predicate on I_F

Properties of I_F: Induction

I_F

φ unary predicate on I_F
Want: If
then $\forall i \in \mathsf{I}_\mathsf{F}.\ \varphi\, i$

Properties of $\mathsf{I_F}$: Induction

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \\
\mathsf{I_F} \arrow[u, "\mathsf{dtor}"]
\end{tikzcd}$$

φ unary predicate on $\mathsf{I_F}$

Want: If

then $\forall i \in \mathsf{I_F}.\ \varphi\ i$

Properties of $\mathsf{I_F}$: Induction

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} \arrow[u, "\mathsf{dtor}"]
\end{tikzcd}$$

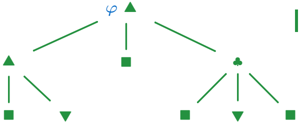
φ unary predicate on $\mathsf{I_F}$

Want: If

then $\forall i \in \mathsf{I_F}.\ \varphi\ i$

Properties of I_F: Induction

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I}_\mathsf{F} \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"] &
\end{tikzcd}$$

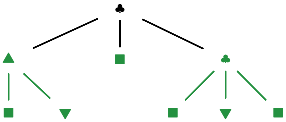


φ unary predicate on I_F

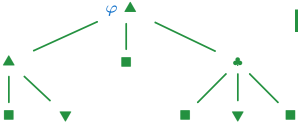
Want: If $\forall i \in \mathsf{I}_\mathsf{F}.$ $\Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I}_\mathsf{F}.\ \varphi\ i$

Properties of $\mathsf{I_F}$: Induction



$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} \arrow[u, "\mathsf{dtor}"] &
\end{tikzcd}$$

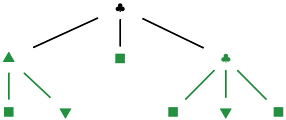


φ unary predicate on $\mathsf{I_F}$

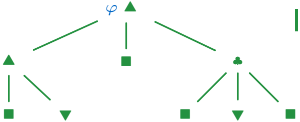
Want: If $\forall i \in \mathsf{I_F}.$ $\Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I_F}.\ \varphi\ i$

Properties of I_{F}: Induction



$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_{\mathsf{F}} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I}_{\mathsf{F}} \\
\mathsf{I}_{\mathsf{F}} \arrow[u, "\mathsf{dtor}"] &
\end{tikzcd}$$

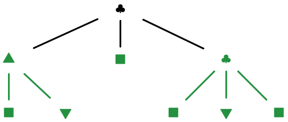


φ unary predicate on I_{F}

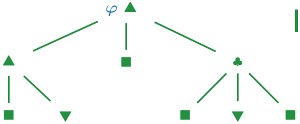
Want: If $\forall i \in \mathsf{I}_{\mathsf{F}}.$ $\Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I}_{\mathsf{F}}.\ \varphi\ i$

Properties of I_{F}: Induction



$$\begin{tikzcd} \mathsf{F}\ \mathsf{I}_{\mathsf{F}} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I}_{\mathsf{F}} \\ \mathsf{I}_{\mathsf{F}} \arrow[u, "\mathsf{dtor}"] & \end{tikzcd}$$

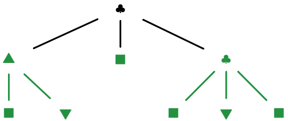


φ unary predicate on I_{F}

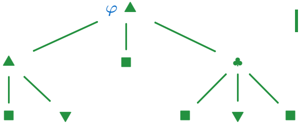
If $\forall i \in \mathsf{I}_{\mathsf{F}}.\ (\forall i' \in \mathsf{Fatoms}\ (\mathsf{dtor}\ i).\ \varphi\ i') \Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I}_{\mathsf{F}}.\ \varphi\ i$

Properties of I_F: Induction



$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I}_\mathsf{F} \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"] &
\end{tikzcd}$$

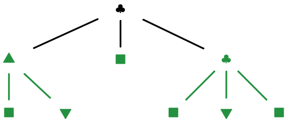


φ unary predicate on I_F

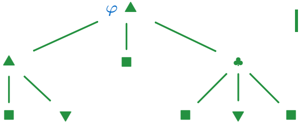
If $\forall i \in \mathsf{I}_\mathsf{F}.\ (\forall i' \in \mathsf{Fatoms}\ (\mathsf{dtor}\ i).\ \varphi\ i') \Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I}_\mathsf{F}.\ \varphi\ i$

Properties of I_F: Induction



$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_\mathsf{F} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I}_\mathsf{F} \\
\mathsf{I}_\mathsf{F} \arrow[u, "\mathsf{dtor}"] \arrow[ur, "\mathsf{components}"'] &
\end{tikzcd}$$

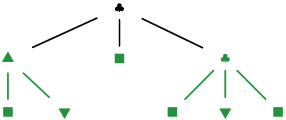


φ unary predicate on I_F

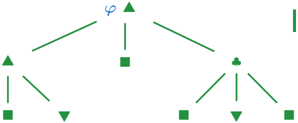
If $\forall i \in \mathsf{I}_\mathsf{F}.\ (\forall i' \in \mathsf{components}\ i.\ \varphi\ i') \Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I}_\mathsf{F}.\ \varphi\ i$

Properties of $\mathsf{I_F}$: Destructor-Style Induction



$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{Fatoms}"] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} \arrow[u, "\mathsf{dtor}"] \arrow[ur, "\mathsf{components}"'] &
\end{tikzcd}$$

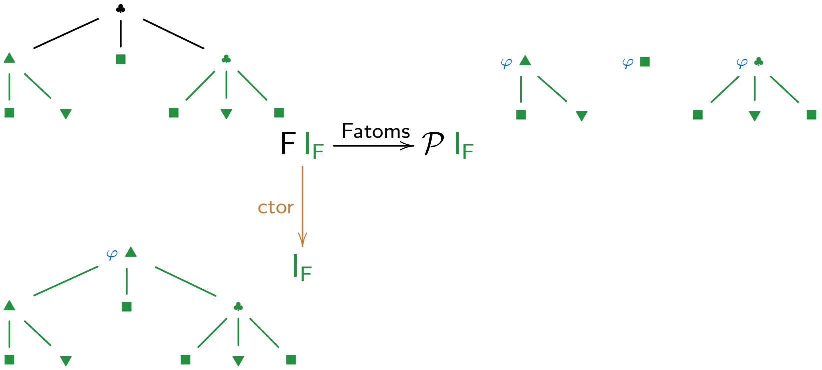


φ unary predicate on $\mathsf{I_F}$

If $\forall i \in \mathsf{I_F}.\ (\forall i' \in \mathsf{components}\ i.\ \varphi\ i') \Rightarrow \varphi\ i$

then $\forall i \in \mathsf{I_F}.\ \varphi\ i$

Properties of I_{F}: Constructor-Style Induction

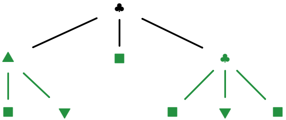


$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I}_{\mathsf{F}} \arrow[r, "\mathsf{Fatoms}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I}_{\mathsf{F}} \\
\mathsf{I}_{\mathsf{F}} &
\end{tikzcd}$$

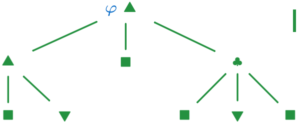
φ unary predicate on I_{F}

If $\forall i \in \mathsf{I}_{\mathsf{F}}.\ (\forall i' \in \textsf{components}\ i.\ \varphi\ i') \Rightarrow \varphi\ i$
then $\forall i \in \mathsf{I}_{\mathsf{F}}.\ \varphi\ i$

Properties of $\mathsf{I_F}$: Constructor-Style Induction



$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{Fatoms}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$



φ unary predicate on $\mathsf{I_F}$

If $\forall x \in \mathsf{F}\ \mathsf{I_F}.\ (\forall i \in \mathsf{Fatoms}\ x.\ \varphi\ i) \Rightarrow \varphi\ (\mathsf{ctor}\ x)$

then $\forall i \in \mathsf{I_F}.\ \varphi\ i$

Bottom line for I_F

Given a natural functor F, $(I_F,\ \mathrm{ctor} : F\, I_F \to I_F)$ satisfies:

Bottom line for I_F

Given a natural functor F, $(\mathsf{I}_\mathsf{F},\ \mathsf{ctor} : \mathsf{F}\ \mathsf{I}_\mathsf{F} \to \mathsf{I}_\mathsf{F})$ satisfies:

ctor bijection

Bottom line for $\mathsf{I_F}$

Given a natural functor F, $(\mathsf{I_F},\ \mathsf{ctor} : \mathsf{F}\ \mathsf{I_F} \to \mathsf{I_F})$ satisfies:

ctor bijection

Iteration (Initial Algebra Property): For all $(A, s : \mathsf{F}\ A \to A)$, there exists a unique function iter_s such that

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{F}\ \mathsf{iter}_s"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"] \\
\mathsf{I_F} \arrow[r, "\mathsf{iter}_s"'] & A
\end{tikzcd}$$

Bottom line for $\mathsf{I_F}$

Given a natural functor F, $(\mathsf{I_F},\ \mathsf{ctor} : \mathsf{F}\ \mathsf{I_F} \to \mathsf{I_F})$ satisfies:

ctor bijection

Iteration (Initial Algebra Property): For all $(A, s : \mathsf{F}\ A \to A)$, there exists a unique function iter_s such that

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{F}\ \mathsf{iter}_s"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"] \\
\mathsf{I_F} \arrow[r, "\mathsf{iter}_s"'] & A
\end{tikzcd}$$

Induction: Given any predicate φ on $\mathsf{I_F}$

$$\frac{\forall x \in \mathsf{F}\ \mathsf{I_F}.\ (\forall \mathsf{i} \in \mathsf{Fatoms}\ \mathsf{x}.\ \varphi\ \mathsf{i}) \Rightarrow \varphi\ (\mathsf{ctor}\ \mathsf{x})}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Bottom line for $\mathsf{I_F}$

Given a natural functor F, $(\mathsf{I_F},\ \mathsf{ctor} : \mathsf{F}\ \mathsf{I_F} \to \mathsf{I_F})$ satisfies:

ctor bijection

$$\boxed{\mathsf{I_F} = \text{the datatype of F}}$$

Iteration (Initial Algebra Property): For all $(A, s : \mathsf{F}\ A \to A)$, there exists a unique function iter_s such that

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{F}\ \mathsf{iter}_s"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"'] \\
\mathsf{I_F} \arrow[r, "\mathsf{iter}_s"'] & A
\end{tikzcd}$$

Induction: Given any predicate φ on $\mathsf{I_F}$

$$\frac{\forall x \in \mathsf{F}\ \mathsf{I_F}.\ (\forall \mathsf{i} \in \mathsf{Fatoms}\ \mathsf{x}.\ \varphi\ \mathsf{i}) \Rightarrow \varphi\ (\mathsf{ctor}\ \mathsf{x})}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype

Let B be a fixed set. $\mathsf{F}\,A = \{*\} + B \times A$

Example of Datatype

Let B be a fixed set. F $A = \{*\} + B \times A$

The shapes of F: Left $*$

Example of Datatype

Let B be a fixed set. $\mathsf{F}\ A = \{*\} + B \times A$

The shapes of F: Left $*$ Right $(b, \_)$ for each $b \in B$

Example of Datatype

Let B be a fixed set. F $A = \{*\} + B \times A$

The shapes of F: Left $*$ Right $(b, \_)$ for each $b \in B$

Or, graphically: $\blacksquare_*$

for each $b \in B$

Example of Datatype

Let B be a fixed set. $\quad \mathsf{F}\ A = \{*\} + B \times A$

The shapes of F: Left $*$ Right $(b, \_)$ for each $b \in B$

Or, graphically:

for each $b \in B$

Who is $\mathsf{I_F}$?

Example of Datatype

Let B be a fixed set. $\quad \mathsf{F}\,A = \{*\} + B \times A$

The shapes of F: Left $*$ $\quad$ Right $(b, \_)$ for each $b \in B$

Or, graphically:

for each $b \in B$

Who is $\mathsf{I_F}$?
Its elements have the form
Right$(b_1, \ldots,$ Right$(b_n,$ Right (Left $*$)$) \ldots)$

Example of Datatype

Let B be a fixed set. $\quad \mathsf{F}\,A = \{*\} + B \times A$

The shapes of F: Left $*$ Right $(b, \_)$ for each $b \in B$

Or, graphically:

for each $b \in B$

Who is I_F?

Its elements have the form

Right$(b_1, \ldots,$ Right$(b_n,$ Right (Left $*$)) $\ldots)$

I.e., essentially lists $b_1 \cdot \ldots \cdot b_n$

Example of Datatype

Let B be a fixed set. F $A = \{*\} + B \times A$

The shapes of F: Left $*$ Right $(b, \_)$ for each $b \in B$

Or, graphically:

for each $b \in B$

Who is I_F?

Its elements have the form

Right$(b_1, \ldots,$ Right$(b_n,$ Right (Left $*$)) $\ldots)$

I.e., essentially lists $b_1 \cdot \ldots \cdot b_n$

So $\mathsf{I}_\mathsf{F} = \mathsf{List}_\mathsf{B}$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad f = \mathsf{iter}_s \quad \mathsf{I_F} = \mathsf{List_B}$

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "f"] \arrow[d, "\mathsf{ctor}"'] & \mathsf{F}\ A \arrow[d, "s"] \\
\mathsf{I_F} \arrow[r, "f"'] & A
\end{tikzcd}$$

$\forall x \in \mathsf{F}\ \mathsf{I_F}.\ \mathsf{f}\ (\mathsf{ctor}\ \mathsf{x}) = \mathsf{s}\ ((\mathsf{F}\ \mathsf{f})\ \mathsf{x})$

Example of Datatype: List

B fixed $\quad$ $\mathsf{F}\ A = \{*\} + B \times A$ $\quad$ $f = \mathsf{iter}_s$ $\quad$ $\mathsf{I_F} = \mathsf{List_B}$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\{*\}+B\times f"] \arrow[d, "\mathsf{ctor}"'] & \{*\} + B \times A \arrow[d, "s"] \\
\mathsf{I_F} \arrow[r, "f"'] & A
\end{tikzcd}$$

$\forall x \in \mathsf{F}\ \mathsf{I_F}.\ \mathsf{f}\ (\mathsf{ctor}\ \mathsf{x}) = \mathsf{s}\ ((\mathsf{F}\ \mathsf{f})\ \mathsf{x})$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad f = \mathsf{iter}_s \quad \mathsf{I_F} = \mathsf{List_B}$

Define:
$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$
$\mathsf{Nil}^A = s\ (\mathsf{Left}\ *) \quad \mathsf{Cons}^A(b, a) = s\ (\mathsf{Right}\ (b, a))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\{*\}+B\times f"] \arrow[d, "\mathsf{ctor}"'] & \{*\} + B \times A \arrow[d, "s"] \\
\mathsf{I_F} \arrow[r, "f"'] & A
\end{tikzcd}$$

$\forall x \in \mathsf{F}\ \mathsf{I_F}.\ \mathsf{f}\ (\mathsf{ctor}\ \mathsf{x}) = \mathsf{s}\ ((\mathsf{F}\ \mathsf{f})\ \mathsf{x})$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad f = \mathsf{iter}_s \quad \mathsf{I_F} = \mathsf{List_B}$

Define:

$$\begin{array}{ll} \mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) & \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i)) \\ \mathsf{Nil}^A = s\ (\mathsf{Left}\ *) & \mathsf{Cons}^A(b, a) = s\ (\mathsf{Right}\ (b, a)) \end{array}$$

$$\begin{tikzcd}
B \times \mathsf{I_F} \arrow[r, "B\times f"] \arrow[d, "\mathsf{Cons}"'] & B \times A \arrow[d, "\mathsf{Cons}^A"] \\
\mathsf{Nil} \in \mathsf{I_F} \arrow[r, "f"'] & A \ni \mathsf{Nil}^A
\end{tikzcd}$$

$\forall x \in \mathsf{F}\ \mathsf{I_F}.\ \mathsf{f}\ (\mathsf{ctor}\ \mathsf{x}) = \mathsf{s}\ ((\mathsf{F}\ \mathsf{f})\ \mathsf{x})$

Example of Datatype: List

B fixed $\quad \mathsf{F}\,A = \{*\} + B \times A \quad f = \mathsf{iter}_s \quad \mathsf{I_F} = \mathsf{List_B}$

Define: $\begin{array}{ll} \mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) & \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i)) \\ \mathsf{Nil}^A = s\ (\mathsf{Left}\ *) & \mathsf{Cons}^A(b, a) = s\ (\mathsf{Right}\ (b, a)) \end{array}$

$$\begin{tikzcd}
B \times \mathsf{I_F} \arrow[r, "B\times f"] \arrow[d, "\mathsf{Cons}"'] & B \times A \arrow[d, "\mathsf{Cons}^A"] \\
\mathsf{Nil} \in \mathsf{I_F} \arrow[r, "f"'] & A \ni \mathsf{Nil}^A
\end{tikzcd}$$

$f\ \mathsf{Nil} = \mathsf{Nil}^A$

$\forall b \in B,\ i \in \mathsf{I_F}.\ \mathsf{f}\ (\mathsf{Cons}\ (\mathsf{b}, \mathsf{i})) = \mathsf{Cons}^{\mathsf{A}}\ (\mathsf{b}, \mathsf{f}\ \mathsf{i})$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad f = \mathsf{iter}_s \quad \mathsf{I_F} = \mathsf{List_B}$

Define: $\quad \mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$
$\quad \mathsf{Nil}^A = s\ (\mathsf{Left}\ *) \quad \mathsf{Cons}^A(b, a) = s\ (\mathsf{Right}\ (b, a))$

$$\begin{tikzcd}
B \times \mathsf{I_F} \arrow[r, "B\times f"] \arrow[d, "\mathsf{Cons}"'] & B \times A \arrow[d, "\mathsf{Cons}^A"] \\
\mathsf{Nil} \in \mathsf{I_F} \arrow[r, "f"'] & A \ni \mathsf{Nil}^A
\end{tikzcd}$$

$f\ \mathsf{Nil} = \mathsf{Nil}^A$

$\boxed{\text{We obtain standard list iteration!}}$

$\forall b \in B,\ i \in \mathsf{I_F}.\ \mathsf{f}\ (\mathsf{Cons}\ (\mathsf{b}, \mathsf{i})) = \mathsf{Cons}^{\mathsf{A}}\ (\mathsf{b}, \mathsf{f}\ \mathsf{i})$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$$\begin{tikzcd}
\mathsf{F}\ \mathsf{I_F} \arrow[r, "\mathsf{Fatoms}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$

$$\frac{\forall x \in \mathsf{F}\ \mathsf{I_F}.\ (\forall \mathsf{i} \in \mathsf{Fatoms}\ \mathsf{x}.\ \varphi\ \mathsf{i}) \Rightarrow \varphi\ (\mathsf{ctor}\ \mathsf{x})}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[rrr, "\text{Left } * \mapsto \varnothing,\ \text{Right } (b,i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & & & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} & & &
\end{tikzcd}$$

$$\frac{\forall x \in \mathsf{F}\ \mathsf{I_F}.\ (\forall \mathsf{i} \in \mathsf{Fatoms}\ \mathsf{x}.\ \varphi\ \mathsf{i}) \Rightarrow \varphi\ (\mathsf{ctor}\ \mathsf{x})}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$

$$\frac{\forall x \in \mathsf{F}\ \mathsf{I_F}.\ (\forall \mathsf{i} \in \mathsf{Fatoms}\ \mathsf{x}.\ \varphi\ \mathsf{i}) \Rightarrow \varphi\ (\mathsf{ctor}\ \mathsf{x})}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$

$$\frac{\begin{array}{l}(\forall i \in \mathsf{Fatoms}\ (\mathsf{Left}\ *).\ \varphi\ i) \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Left}\ *)) \\ \forall b \in B,\ i \in \mathsf{I_F}.\ (\forall \mathsf{i}' \in \mathsf{Fatoms}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})).\ \varphi\ \mathsf{i}') \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[rr, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} & &
\end{tikzcd}$$

$$\frac{\begin{array}{c}(\forall i \in \varnothing.\ \varphi\ i) \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Left}\ *)) \\ \forall b \in B,\ i \in \mathsf{I_F}.\ (\forall \mathsf{i'} \in \mathsf{Fatoms}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})).\ \varphi\ \mathsf{i'}) \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$

$$\frac{\begin{array}{l}\varphi\ (\mathsf{ctor}\ (\mathsf{Left}\ *)) \\ \forall b \in B,\ i \in \mathsf{I_F}.\ (\forall \mathsf{i}' \in \mathsf{Fatoms}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})).\ \varphi\ \mathsf{i}') \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[rr, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b, i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} & &
\end{tikzcd}$$

$$\frac{\varphi\ \mathsf{Nil} \qquad \forall b \in B,\ i \in \mathsf{I_F}.\ (\forall \mathsf{i'} \in \mathsf{Fatoms}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})).\ \varphi\ \mathsf{i'}) \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})))}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[rr, "\mathsf{Left}\ * \mapsto \varnothing,\ \ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} & &
\end{tikzcd}$$

$$\frac{\begin{array}{l}\varphi\ \mathsf{Nil} \\ \forall b \in B,\ i \in \mathsf{I_F}.\ (\forall \mathsf{i'} \in \{\mathsf{i}\}.\ \varphi\ \mathsf{i'}) \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$

$$\frac{\begin{array}{l}\varphi\ \mathsf{Nil} \\ \forall b \in B,\ i \in \mathsf{I_F}.\ \varphi\ \mathsf{i} \Rightarrow \varphi\ (\mathsf{ctor}\ (\mathsf{Right}\ (\mathsf{b}, \mathsf{i})))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[rr, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} & &
\end{tikzcd}$$

$$\frac{\begin{array}{l}\varphi\ \mathsf{Nil} \\ \forall b \in B,\ i \in \mathsf{I_F}.\ \varphi\ \mathsf{i} \Rightarrow \varphi\ (\mathsf{Cons}\ (\mathsf{b}, \mathsf{i}))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

Example of Datatype: List

B fixed $\quad \mathsf{F}\ A = \{*\} + B \times A \quad \mathsf{I_F} = \mathsf{List_B}$

$\mathsf{Nil} = \mathsf{ctor}\ (\mathsf{Left}\ *) \quad \mathsf{Cons}(b, i) = \mathsf{ctor}\ (\mathsf{Right}\ (b, i))$

$$\begin{tikzcd}
\{*\} + B \times \mathsf{I_F} \arrow[r, "\mathsf{Left}\ * \mapsto \varnothing,\ \mathsf{Right}\ (b{,}i) \mapsto \{i\}"] \arrow[d, "\mathsf{ctor}"'] & \mathcal{P}\ \mathsf{I_F} \\
\mathsf{I_F} &
\end{tikzcd}$$

Obtain standard list induction!

$$\frac{\begin{array}{l}\varphi\ \mathsf{Nil} \\ \forall b \in B,\ i \in \mathsf{I_F}.\ \varphi\ \mathsf{i} \Rightarrow \varphi\ (\mathsf{Cons}\ (\mathsf{b}, \mathsf{i}))\end{array}}{\forall i \in \mathsf{I_F}.\ \varphi\ \mathsf{i}}$$

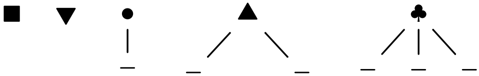
Iterating Shape Composition Revisited

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

Iterating Shape Composition Revisited

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

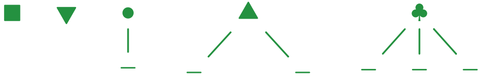
The shapes of F:



Iterating Shape Composition Revisited

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

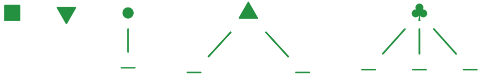
Copies of the shapes of F:



Iterating Shape Composition Revisited

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:

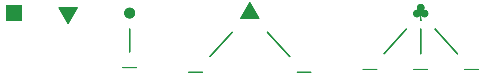


Put them together by plugging in shape for content slot

Iterating Shape Composition Revisited

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:

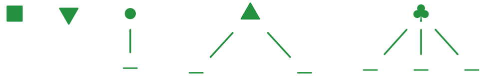


Put them together by plugging in shape for content slot

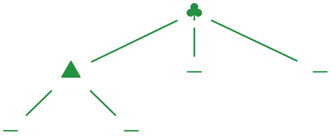
Iterating Shape Composition Revisited

Natural functor F : Set → Set

Copies of the shapes of F:



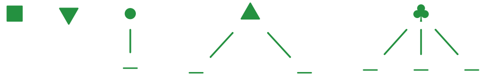
Put them together by plugging in shape for content slot



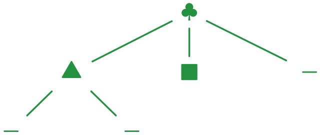
Iterating Shape Composition Revisited

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



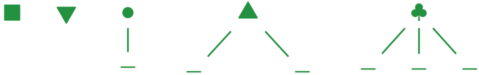
Put them together by plugging in shape for content slot



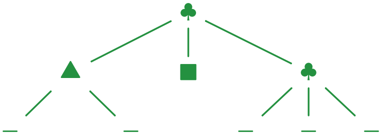
Iterating Shape Composition Revisited

Natural functor $F : \mathrm{Set} \to \mathrm{Set}$

Copies of the shapes of F:



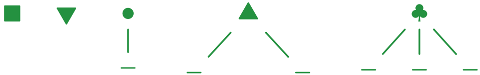
Put them together by plugging in shape for content slot



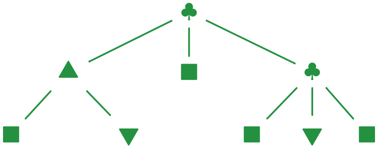
Iterating Shape Composition Revisited

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



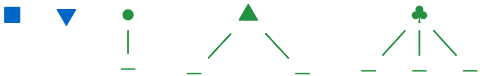
Put them together by plugging in shape for content slot until there are no lingering slots left!



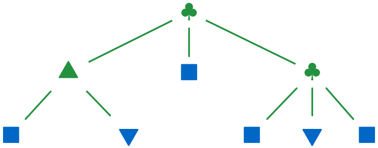
Iterating Shape Composition Revisited

Natural functor $\mathsf{F} : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



Put them together by plugging in shape for content slot
until there are no lingering slots left!

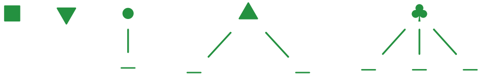


The leaves are always empty-content shapes

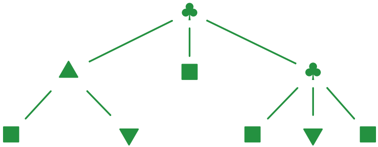
Iterating Shape Composition Revisited

Natural functor F : Set → Set

Copies of the shapes of F:



Put them together by plugging in shape for content slot until there are no lingering slots left!

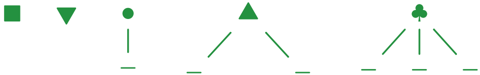


The leaves are always empty-content shapes

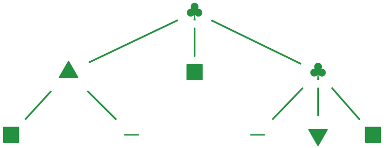
Iterating Shape Composition Revisited

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



Put them together by plugging in shape for content slot
until there are no lingering slots left!

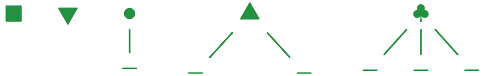


Allow infinite couplings

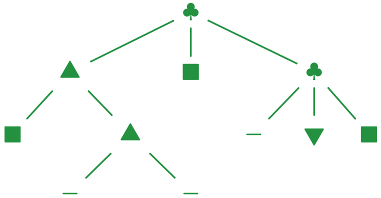
Iterating Shape Composition Revisited

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



Put them together by plugging in shape for content slot
until there are no lingering slots left!

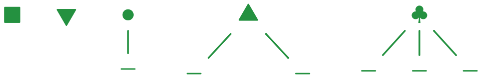


Allow infinite couplings

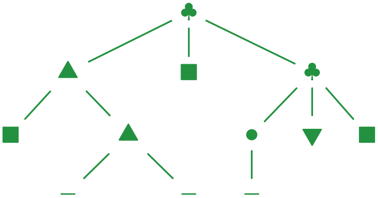
Iterating Shape Composition Revisited

Natural functor F : Set → Set

Copies of the shapes of F:



Put them together by plugging in shape for content slot until there are no lingering slots left!

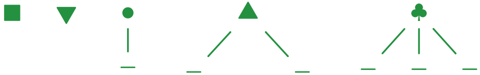


Allow infinite couplings

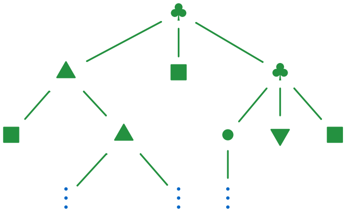
Iterating Shape Composition Revisited

Natural functor F : Set → Set

Copies of the shapes of F:



Put them together by plugging in shape for content slot
until there are no lingering slots left!

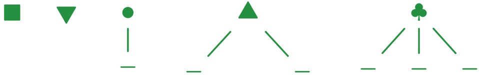


Allow infinite couplings

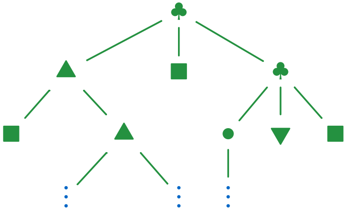
Iterating Shape Composition Revisited

Natural functor $F : \mathsf{Set} \to \mathsf{Set}$

Copies of the shapes of F:



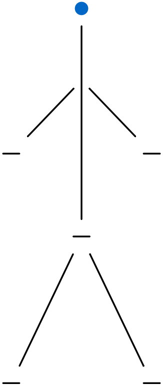
Put them together by plugging in shape for content slot until there are no lingering slots left!



Define J_F = the set of all such (possibly) infinitary couplings

Welcome to Codatatypes

End of Part I



Many thanks for your attention
See you in 30 minutes