BME TMIT
2022

14/7

Németh Gábor

Funkcionális programozás C++-ban

Funktorok

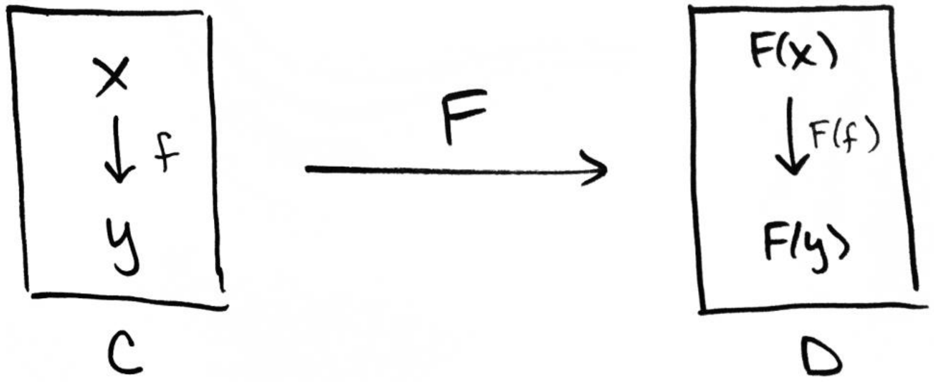
Funktorok és C++ függvényobjektumok

- funktor ≠ C++ függvény objektum
  - sajnos sokan zsargonként használják a „funktor” főnevet C++ függvényobjektumok azonosítására
  - ne tegyük!

Funktorok

- a funktor leképzés kategóriák között, amely megőrzi a struktúrát

- az $F: \boldsymbol{C} \to \boldsymbol{D}$ funktor a $\boldsymbol{C}$ és $\boldsymbol{D}$ kategória között,
  - ha a $\boldsymbol{C}$ kategória valamennyi x objektumához létezik $F(x) \in \boldsymbol{D}$ objektum
  - ha a $\boldsymbol{C}$ kategória valamennyi $x \xrightarrow{f} y$ morfizmusához létezik $\mathrm{F}(x) \xrightarrow{F(f)} F(y)$ morfizmus $\boldsymbol{D}$-ben
  - illetve, ha
    - F-re érvényesek a kompozíció szabályai, azaz $F(g \circ f) = F(g) \circ F(f)$ $\boldsymbol{D}$-ben, ha g és f komponálható morfizusok $\boldsymbol{C}$-ben
    - F az identitásmorfizmusokat identitásmorfizmusokba képezi, azaz $F(id_x) = id_{F(x)}$ $\boldsymbol{C}$ valamennyi x objektumára.

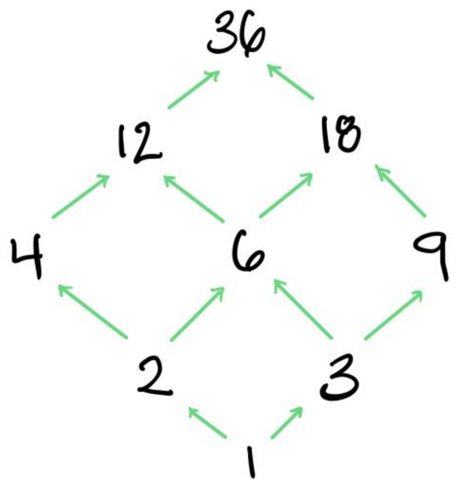


Funktortörvények

- Legyen $\boldsymbol{C}$ és $\boldsymbol{D}$ két kategória
- $I.$
$$\forall o \in ob(\boldsymbol{C})\ \exists\ F(o) \in ob(\boldsymbol{D})$$
- $II.$
$$\forall o \in ob(\boldsymbol{C}), F\ (id_o) = id_{F(o)},\ \text{ahol}\ id_{F(o)} \in mor(\boldsymbol{D})$$
- $III.$
$$\forall t, x, y \in mor(\boldsymbol{C}), \text{ahol}\ t = y \circ x, F(t) = F(y) \circ F(x)$$

A Fibonacci sorozat funktor I.

- A természetes számok az oszthatósággal mint morfizmussal kategória.



A Fibonacci sorozat funktor II.

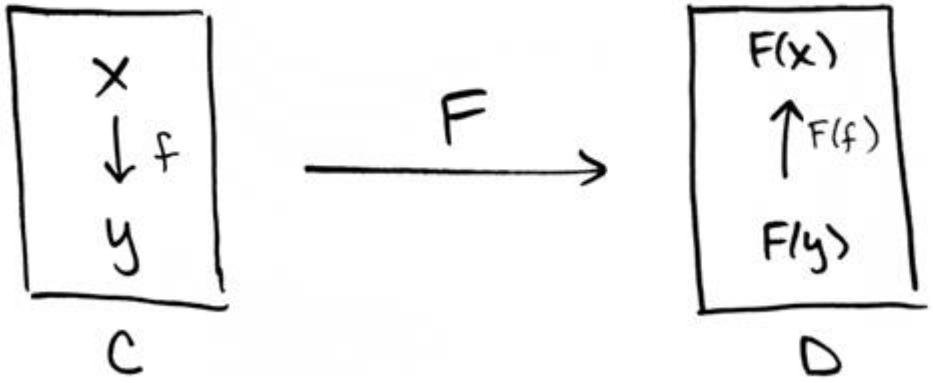
- legyen $F\colon \mathbb{N} \to \mathbb{N}$ egy függvény amely az n természetes számhoz hozzárendeli a n-edik Fibonacci számot, azaz $F(n) := F_n$
- ekkor $n|m$-ből következik, hogy $F_n|F_m, \forall\, n, m \geq 1$

- objektumok összerendelése: $n \to F_n$
- morfizusok megfeltetése:

$$\begin{tikzcd}
6 \arrow[d, "\text{divides}"'] & \overset{F}{\longmapsto} & F_6 = 8 \arrow[d, "\text{divides}"] \\
36 & & F_{36} = 14{,}930{,}352
\end{tikzcd}$$

Speciális funktorok

- endo-funktor
  - ha $\boldsymbol{C}$ és $\boldsymbol{D}$ megegyezik, azaz F $\boldsymbol{C}$-ből $\boldsymbol{C}$-be képez, akkor F az egy endo-funktor
  - programozásban endo-funktorokkal dolgozunk, hiszen a típusrendszerünk maga egy kategória, és azon a rendszeren belül tudunk csak mozogni
- kovariáns funktor
  - simán csak funktorként hivatkozunk rá
- kontravariáns funktor
  - úgy képzi le a morfizmusokat, hogy azok iránya megfordul



ADT és funktorok

ADT
Maybe funktor

ADT-k és funktorok

- Minden algebrai adattípus funktoriális
  - nem bizonyítjuk

A Maybe funktor

- Maybe a = 1 + a

```
using Nothing = std::monostate;
template<typename T>
using Maybe = std::variant<Nothing,T>;
```

- típus → típus

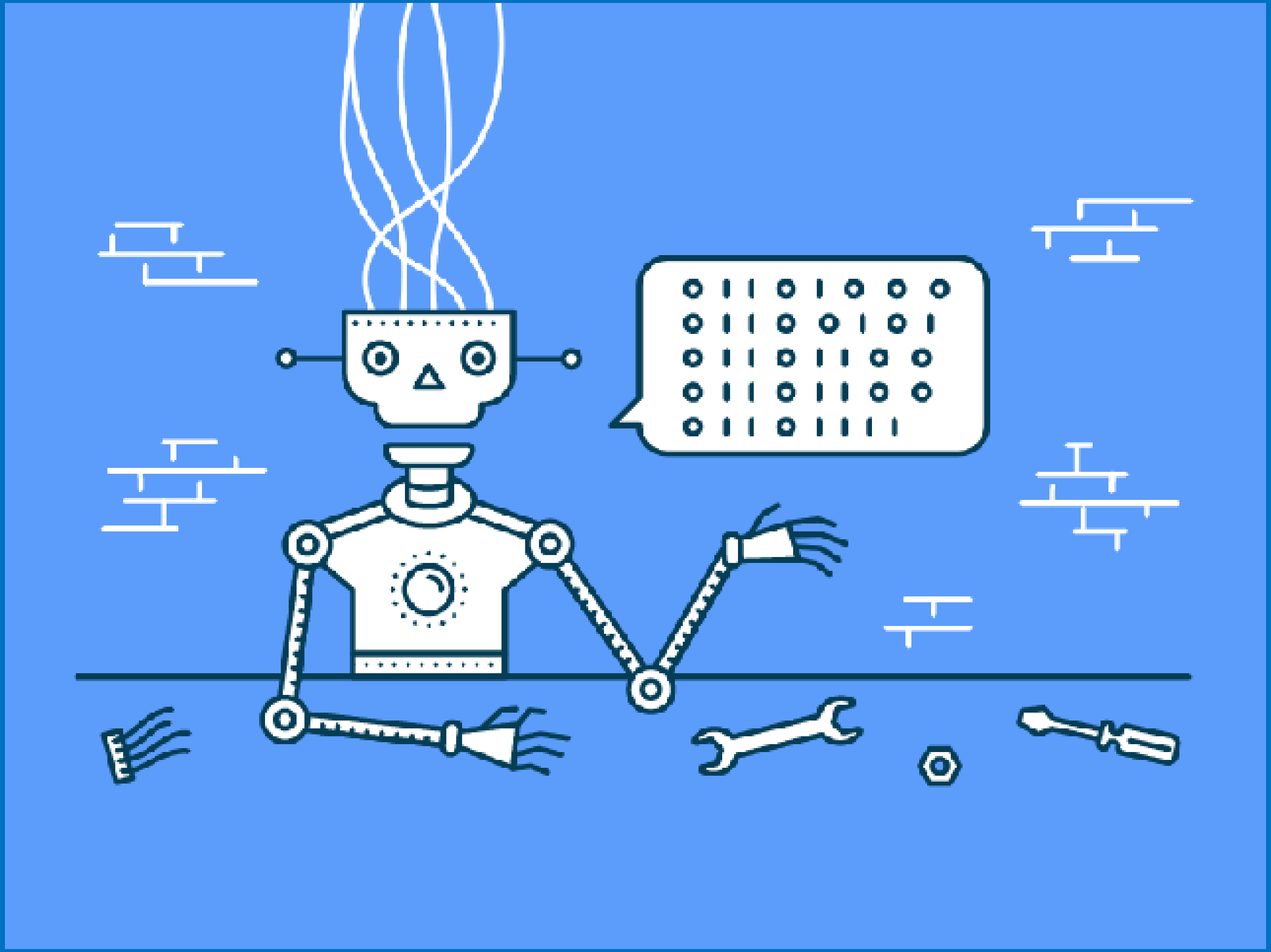
$$T \to \text{Maybe}<T>$$

- függvény → függvény

$$int\ \text{add}_5\ (int);\ \ \to\ \ \text{Maybe}<int>\ \text{add}_5(\text{Maybe}<int>);$$

  - megjegyzés: gyakorlatban ezt *fmap*-nak hívják, pl. $(fmap\ add\_5)\ Nothing\ \ ==\ \ Nothing$

ex_0: Maybe funktor



A Maybe funktor?

- Egy tetszőleges függvényre
  - ① $(fmap\ f)(just(x)) = just(f(x))$
  - ② $(fmap\ f)(Nothing) = Nothing$
- és
  - ③ $id\ o = o, \forall o \in ob(C)$

- F teljesíti a *funktor törvényeket*
  - *I.* minden típust be tudunk „csomagolni” *Maybe*-be, triviális
  - *II.* az identitás megmarad

$$fmap\ id\ Nothing = \ \dots = id\ Nothing$$

$$fmap\ id\ Nothing \xrightarrow{2} Nothing = \ \dots \ = id\ Nothing$$

$$fmap\ id\ Nothing \xrightarrow{2} Nothing = \ Nothing \xleftarrow{3} id\ Nothing$$

$Nothing$

$$fmap\ id\ just(x) \ = \ \dots \ = \ id\ just(x)$$

$$fmap\ id\ just(x) \xrightarrow{1} just(id(x)) = \ \dots \ = \ id\ just(x)$$

$$fmap\ id\ just(x) \xrightarrow{1} just(id(x)) \xrightarrow{3} just(x) = just(x) \xleftarrow{3} id\ just(x)$$

$just(x)$

A Maybe funktor?

- F teljesíti a *funktor törvényeket (folyt.)*
  - *III.* a kompozíció megmarad

Nothing

$$fmap\ (g \circ f)\ Nothing = \ \ldots\ = (fmap\ g) \circ (fmap\ f)\ Nothing$$

$$fmap\ (g \circ f)\ Nothing \xrightarrow{2} Nothing = \ \ldots\ = (fmap\ g) \circ (fmap\ f)\ \ Nothing$$

$$fmap\ (g \circ f)\ Nothing \xrightarrow{2} Nothing = \ \ldots = (fmap\ g)\big((fmap\ f)(Nothing)\big) \xleftarrow{g \circ f\ x = g(f(x))} (fmap\ g) \circ (fmap\ f)\ Nothing$$

$$fmap\ (g \circ f)\ Nothing \xrightarrow{2} Nothing = \ \ldots = fmap(g)\ Nothing \xleftarrow{2} (fmap\ g)\big((fmap\ f)(Nothing)\big) \xleftarrow{g \circ f\ x = g(f(x))} (fmap\ g) \circ (fmap\ f)\ Nothing$$

$$fmap\ (g \circ f)\ Nothing \xrightarrow{2} Nothing = \quad = Nothing \xleftarrow{2} (fmap\ g)\ Nothing \xleftarrow{2} (fmap\ g)\big((fmap\ f)(Nothing)\big) \xleftarrow{g \circ f\ x = g(f(x))} (fmap\ g) \circ (fmap\ f)\ Nothing$$

$just(x)$

$$fmap\ (g \circ f)\ just(x) = \ \ldots = (fmap\ g) \circ (fmap\ f)\ just(x)$$

$$fmap\ (g \circ f)\ just(x) \xrightarrow{1} just(g \circ f\ x) = \ \ldots = (fmap\ g) \circ (fmap\ f)\ \ just(x)$$

$$fmap\ (g \circ f)\ just(x) \xrightarrow{1} just(g \circ f\ x) \xrightarrow{g \circ f\ x = g(f(x))} just\Big(g\big(f(x)\big)\Big) = \ \ldots = (fmap\ g)\big((fmap\ f) just(x)\big) \xleftarrow{g \circ f\ x = g(f(x))} (fmap\ g) \circ (fmap\ f)\ just(x)$$

$$fmap\ (g \circ f)\ just(x) \xrightarrow{1} just(g \circ f\ x) \xrightarrow{g \circ f\ x = g(f(x))} just\Big(g\big(f(x)\big)\Big) = \ \ldots$$

$$= (fmap\ g)\ \Big(just\big(f(x)\big)\Big) \xleftarrow{1} (fmap\ g)\big((fmap\ f)\ \ just(x)\big) \xleftarrow{g \circ f\ x = g(f(x))} (fmap\ g) \circ (fmap\ f)\ just(x)$$

$$fmap\ (g \circ f)\ just(x) \xrightarrow{1} just(g \circ f\ x) \xrightarrow{g \circ f\ x = g(f(x))} just\Big(g\big(f(x)\big)\Big) =$$

$$= just\Big(g\big(f(x)\big)\Big) = \xleftarrow{1} (fmap\ g)\ \Big(just\big(f(x)\big)\Big) \xleftarrow{1} (fmap\ g)\big((fmap\ f)\ \ just(x)\big) \xleftarrow{g \circ f\ x = g(f(x))} (fmap\ g) \circ (fmap\ f)\ \ just(x)$$

Természetes transzformációk

Természetes transzformációk I.

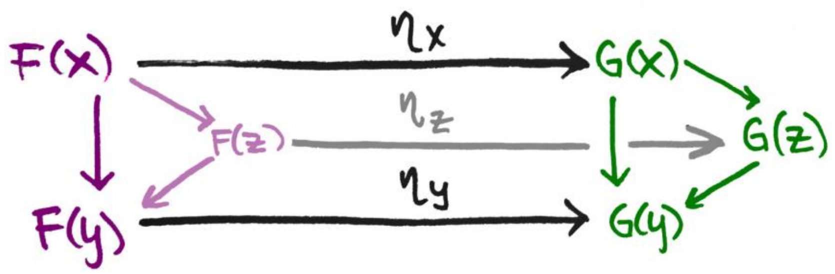
- Legyen F és G két funktor a $\boldsymbol{C}$ és $\boldsymbol{D}$ kategória között. A η: $F \Longrightarrow G$ természetes transzformáció az alábbi tulajdonságokkal rendelkezik
  - létezik a $F(x) \xrightarrow{\eta_x} G(x)$ morfizmus $\boldsymbol{C}$ minden x objektumára
  - ha $x \xrightarrow{f} y$ morfizmus a $\boldsymbol{C}$ kategóriában, akkor $G(f) \circ \eta_x = \eta_y \circ F(f)$, azaz az alábbi diagram kommutatív

$$
\begin{tikzcd}
F(x) \arrow[r, "\eta_x"] \arrow[d, "F(f)"'] & G(x) \arrow[d, "G(f)"] \\
F(y) \arrow[r, "\eta_y"'] & G(y)
\end{tikzcd}
$$

  - vegyük észre, hogy a η természetes transzformáció az összes $\eta_x$morfizmus összessége: $\eta = (\eta_x)_{x \in \boldsymbol{C}}$

Természetes transzformációk II.

- a természetes transzformáció leképezés egy diagramról és másikra
- ezekkel a transzformációkkal kiegészítve a kapott diagram kommutatív



Speciális természetes transzformációk I.

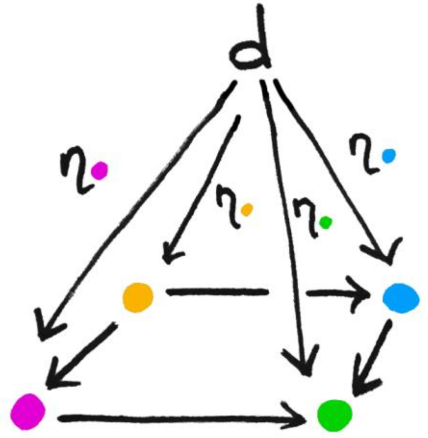
- F és G is konstans
  - Legyenek $F, G\colon \boldsymbol{C} \to \boldsymbol{D}$ konstans funktorok.
    - képezze F a $\boldsymbol{C}$ kategória valamennyi objektumát a $d \in \boldsymbol{D}$ objektumba, míg G képezze a $\boldsymbol{C}$ kategória objektumait a $d' \in \boldsymbol{D}$ objektumba
    - képezze F a $\boldsymbol{C}$ minden morfizmusát a id_d $\boldsymbol{D}$ kategóriabeli morfizmusba
    - képezze G a $\boldsymbol{C}$ minden morfizmusát a $id_{d'}$ $\boldsymbol{D}$ kategóriabeli morfizmusba

  - Ekkor a természetes transzformáció F és G között a $d \overset{\eta}{\to} d'$ morfizmus

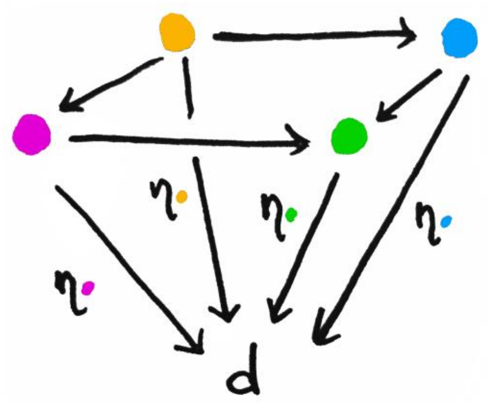
$$d \xrightarrow{\eta} d'$$

Speciális természetes transzformációk II.

- F konstans
  - legyen F konstans valamely $d \in \boldsymbol{D}$ képpel
  - legyen G tetszőleges funktor
  - ekkor $\eta: F \Longrightarrow G$ természetes transzformáció a $d \xrightarrow{\eta_x} G(x)$ $\forall\, x \in \boldsymbol{C}$ leképezések összessége, amelyek teljesítik a $G(f) \circ \eta_x = \eta_y$
- kúp G felett



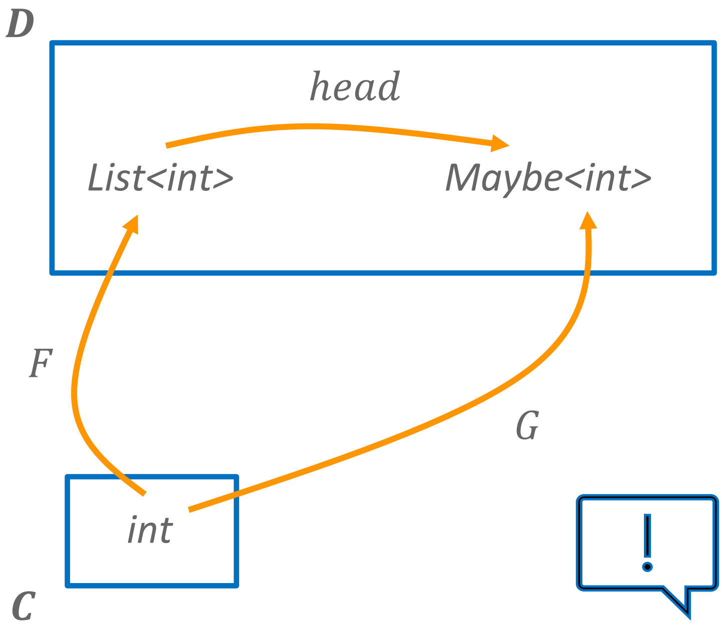
- G konstans
  - legyen G konstans valamely $d \in \boldsymbol{D}$ képpel
  - legyen F tetszőleges funktor
  - ekkor $\eta: F \Longrightarrow G$ természetes transzformáció a $F(x) \xrightarrow{\eta_x} d$ $\forall\, x \in \boldsymbol{C}$ leképezések összessége, amelyek teljesítik a $\eta_y \circ F(x) = \eta_x$
- kúp F alatt



Lista, Maybe, természetes transzformáció

- legyen $head$ egy függvény, amely visszaadja egy lista első elemét, ha az létezik ($just(a)$), egyébként $Nothing$-ot
  - $head: [a] \quad \rightarrow Maybe\ a$
  - $head\ \ [\ ] \quad = Nothing$
  - $head\ \ (x{:}\,xs) = just(x)$

- mutassa meg, hogy a $head$ természetes transzformáció



$$G(f) \circ \eta_x = \eta_y \circ F(f)$$

$(fmap\ f \circ head)\ [] = (fmap\ f)(head\ []) = (fmap\ f)\ Nothing = Nothing$
$(head \circ fmap\ f)\ [] = (head)\big((fmap\ f)[]\big) = head\ [] = Nothing$

$(fmap\ f \circ head)(x{:}\,xs) = (fmap\ f)\big(head\ (x{:}\,xs)\big) = (fmap\ f)(just(x)) = just(f\ x)$
$(head \circ fmap\ f)(x{:}\,xs) = head\ (f(x) : fmap\ f\ xs) = just(f\ x)$

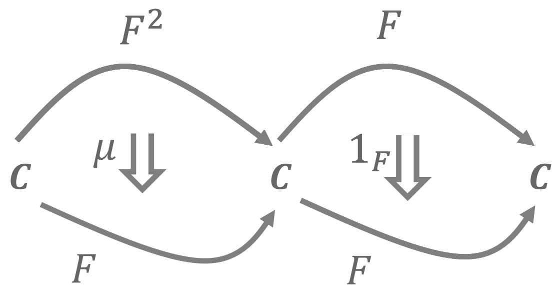
Ha úgy gondolunk a funktorra mint konténerre, akkor a természetes transzformáció módosítja a konténert, de a tartalmát nem.

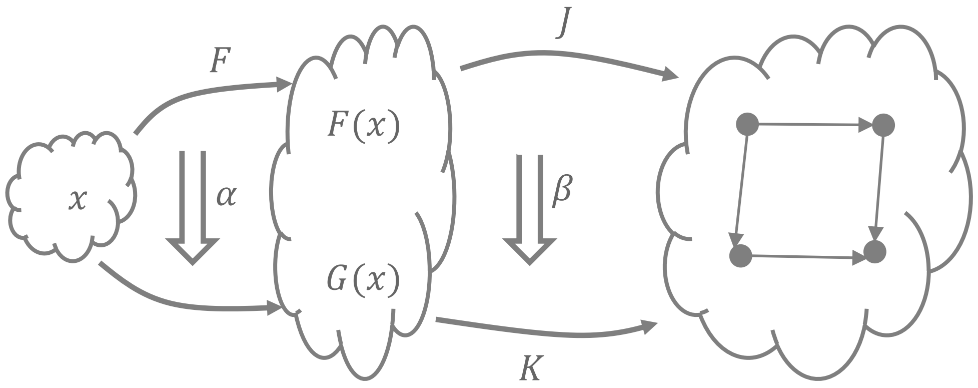
Funktorok kategóriája

Funktorok kategóriája

- a természetes leképezés leképezés funktorok között
- funktorok kategóriája
  - valamennyi $\boldsymbol{C}$ és $\boldsymbol{D}$ párra
  - objektumok: funktorok $\boldsymbol{C}$ és $\boldsymbol{D}$ között
  - morfizmusok: természetes transzformációk a funktorok között
    - identitás természetes transzformáció: 1_F, amelynek az elemei az $id_{F(x)}: F(x) \to F(x)$ morfizmusok

- kompozíció
  - funktor és természetes transzformáció kompozíciójának nincs értelme
  - $F^2 = F \circ F$
  - $F^3 = F \circ F^2$





Monádok

A monád I.

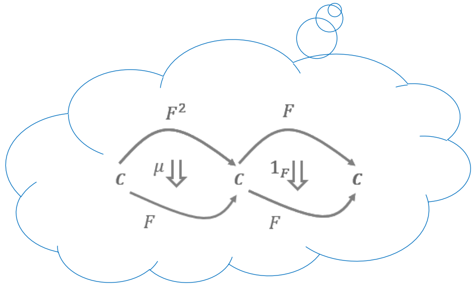
- egy $M = (T, \mu, \eta)$ monád nem más mint egy
  - T endo-funktor
  - $\mu: T^2 \to T$, természetes transzfromáció (*join*), ahol $T^2 = T \circ T$
    - $\mu_a: T(T\ a) \to T\ a, \forall a \in ob(C)$
  - $\eta: I \to T$, természetes transzformáció (*return*)
    - $\eta_a: a \to T\ a, \forall a \in ob(C)$

- ahol az alábbiaknak teljesülnek (***monad-laws***):
  - $\mu \circ 1_T \circ \mu = \mu \circ \mu \circ 1_T$
  - $\mu \circ 1_T \circ \eta = \mu \circ \eta \circ 1_T = 1_T$
  - ahol 1_T az identitás természetes transzformáció

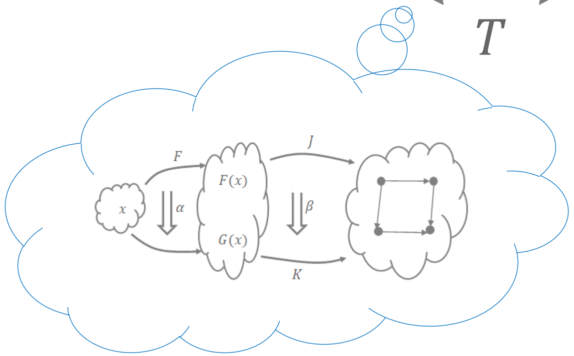
A monád II.

- a monád törvények törvények vizuálisan az endo-fuktorok kategóriájában az alábbi diagramokkal szemléltethetőek

$$
\begin{tikzcd}
T^3 \arrow[r, "\mu \circ 1_T"] \arrow[d, "1_T \circ \mu"'] & T^2 \arrow[d, "\mu"] \\
T^2 \arrow[r, "\mu"'] & T
\end{tikzcd}
$$



$$
\begin{tikzcd}
I \circ T \arrow[dr] & T^2 \arrow[l, "\eta \circ 1_T"'] \arrow[r, "1_T \circ \eta"] \arrow[d, "\mu"] & T \circ I \arrow[dl] \\
& T &
\end{tikzcd}
$$

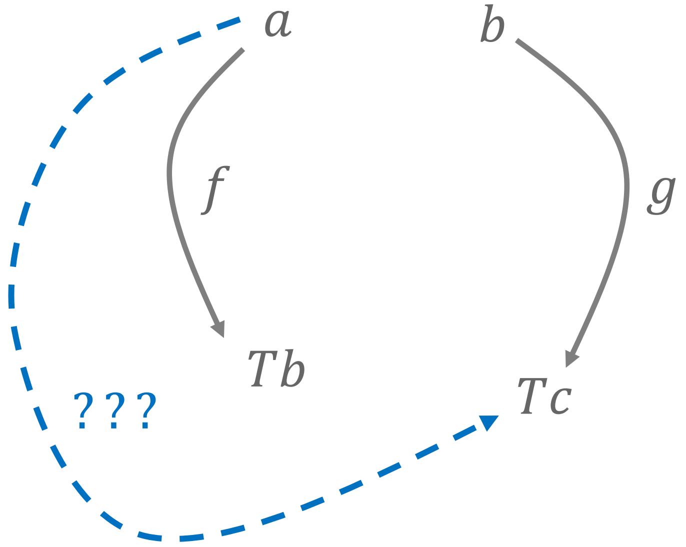


A monád gyakorlati értelmezése

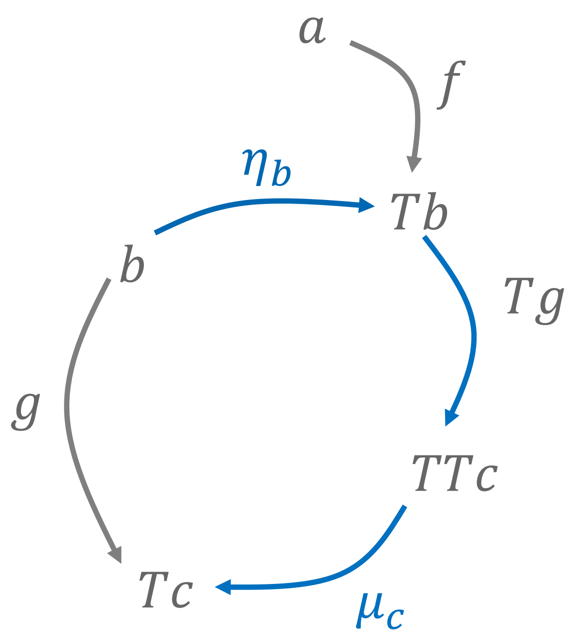
- Mit jelent a gyakorlatban az η (*return*) és μ (*join*):
  - $\eta: 1_C \rightarrow T$ (return): Ha van egy T endo-funktorom , akkor létezik $\eta_x: 1_C x \rightarrow Tx$ morfizmus, $1_C x = x$, azaz $\eta_x: x \rightarrow Tx$
    - ez általában a konstruktort reprezentálja
    - pl.: Maybe<int>(3)
  - $\mu: T^2 \rightarrow T$ (*join*): ehhez nehezebb analógiát rendelni
    - de szemléletesen,tudunk-e
      - Maybe<Maybe<*int*>> -ből Maybe<*int*>-et csinálni
      - List<List<*double*>> -ből List<*double*>-t csinálni
    - és ha igen, hogyan?

Kompozíciók és monádok

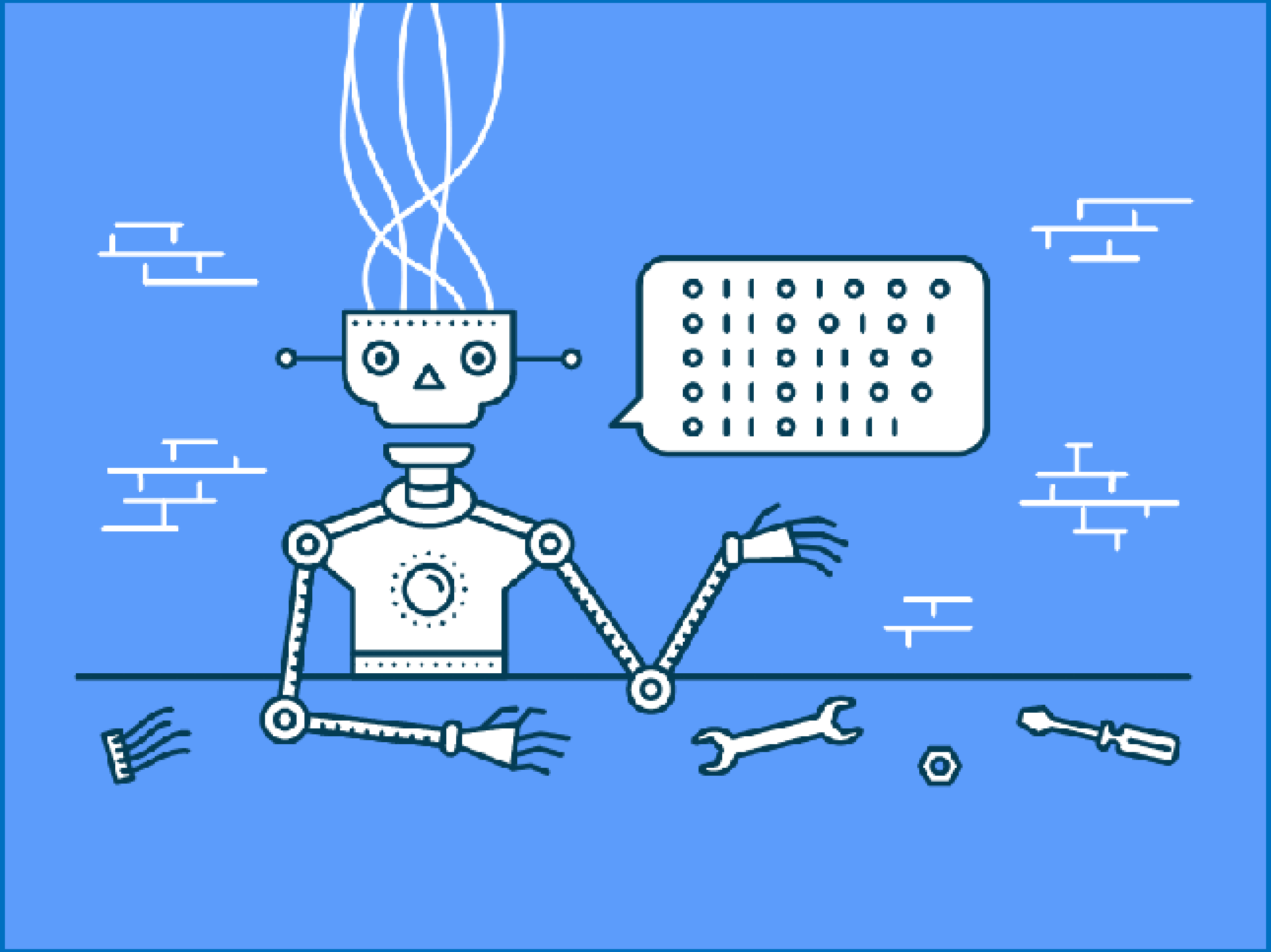
- Minden a kompozícióról szól!



- $M = (T, \mu, \eta)$ monád



ex_1: Monád



Köszönöm a figyelmet!

Folytatjuk…