Seminář 3: Produkty, koprodukty a algebraické datové typy (ADT)

Produkty

$$
\begin{tikzcd}
 & x \arrow[dl, "f"'] \arrow[d, dashed, "q"] \arrow[dr, "g"] & \\
a & a \times b = c \arrow[l, "p_1"] \arrow[r, "p_2"'] & b
\end{tikzcd}
$$

- Objekt $c \in \mathcal{C}$ s projekcemi $p_1 : c \to a$ a $p_2 : c \to b$ je produkt objektů a a $b \iff \forall x \in \mathcal{C}. \forall f : x \to a, g : x \to b. \exists! q : x \to c.\ (f = p_1 \circ q \land g = p_2 \circ q)$
- q *faktorizuje* f a $g \iff f = p_1 \circ q \land g = p_2 \circ q$
- q je *univerzální mapování*

Příklady

- V **Set** *kartézský součin*
- V **Poset** (částečně uspořádaná množina) *infimum*

Haskell

```
data (,) a b = (,) a b
fst (x, _) = x -- p_1
snd (_, y) = y -- p_2

q :: (j -> a) -> (j -> b) -> j -> (,) a b
q f g  x = (f x, g x)
```

Koprodukty (sumy)

$$
\begin{tikzcd}
 & x & \\
a \arrow[ur, "f"] \arrow[r, "i_1"'] & a + b = c \arrow[u, dashed, "q"] & b \arrow[ul, "g"'] \arrow[l, "i_2"]
\end{tikzcd}
$$

- Objekt $c \in \mathcal{C}$ se zahrnutími (*inkluzemi*) $i_1 : c \leftarrow a$ a $i_2 : c \leftarrow b$ je koprodukt objektů a a $b \iff \forall x \in \mathcal{C}. \forall f : x \leftarrow a, g : x \leftarrow b. \exists! q : x \leftarrow c.\ (f = q \circ i_1 \land g = q \circ i_2)$

Příklady

- V **Set** *disjunktní sjednocení*
- V **Poset** (částečně uspořádaná množina) *supremum*

Haskell

```
data Either a b = Left a | Right b
-- i_1 = Left
-- i_2 = Right

q :: (a -> j) -> (b -> j) -> Either a b -> j
q f _ (Left x)  = f x
q _ g (Right x) = g x
```

Bifunktory

Produkt kategorií $\mathcal{C} \times \mathcal{D}$

Skládá se z

- $O_{\mathcal{C}\times\mathcal{D}} = \{(c, d) | c \in O_{\mathcal{C}}, d \in O_{\mathcal{D}}\}$
- $A_{\mathcal{C}\times\mathcal{D}} = \{(f, g) | f : a \to c \in A_{\mathcal{C}}, g : b \to d \in A_{\mathcal{D}}\}$

Musí platit

- *identita* pro (a, b) je (id_a, id_b)
- $(f, h) \circ (g, k) \equiv (f \circ g, h \circ k)$

Bifunktor F = funktor z produktu kategorií $F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$

V každé kategorii $\mathcal{C}$, kde pro každé dva objekty a a b můžemem najít jejich produkt, existuje bifunktor $\times$ (zapisován infixově).

$$\begin{aligned} \times &: \mathcal{C} \times \mathcal{C} \to \mathcal{C} \\ &: (a, b) \mapsto a \times b \\ &: (f : a \to a', g : b \to b') \mapsto (f \times g : a \times b \to a' \times b') \end{aligned}$$

$$\begin{tikzcd} a \arrow[d, "f"'] & a \times b \arrow[l, "p_1"'] \arrow[r, "p_2"] \arrow[d, dashed, "f \times g"] & b \arrow[d, "g"] \\ a' & a' \times b' \arrow[l, "p'_1"] \arrow[r, "p'_2"'] & b' \end{tikzcd}$$

Bifunktory v Haskellu

```haskell
class Bifunctor f where
  bimap :: (a -> b) -> (c -> d) -> f a c -> f b d
  bimap g h = first g . second h
  first :: (a -> b) -> f a c -> f b c
  first g = bimap g id
  second :: (c -> d) -> f a c -> f a d
  second h = bimap id h
  {-# MINIMAL bimap | first, second #-}

instance Bifunctor (,) where
  bimap g h (x, y) = (g x, h y)

instance Bifunctor Either where
  bimap g _ (Left x)  = Left  (g x)
  bimap _ h (Right y) = Right (h y)
```

ADT a funktory

- Pro každý ADT umí GHC instancovat funktor s pomocí rozšíření (od GHC 9.2.1 bez rozšíření)

```haskell
{-# LANGUAGE DerivingFunctor #-}
data Example a = Ex a Int (Example a) (Example Int)
    deriving (Functor)
```