3. Implemente mirror $= (\!|\, g \,|\!)$ em Haskell definindo previamente `outLTree` e o combinador `cataLTree` (catamorfismo de LTrees).

In [1]:

```
-- loading Cp.hs
:opt no-lint
:load ../src/Cp.hs
:set -XNPlusKPatterns
```

In [2]:

```
data LTree a = Leaf a | Fork (LTree a, LTree a) deriving (Show)

:t Leaf
:t Fork

-- outLTree

outLTree (Leaf a) = i1 a
outLTree (Fork (a,b)) = i2 (a,b)

:t outLTree

-- cataLTree

cataLTree g = g . (id -|- (cataLTree g >< cataLTree g)) . outLTree

:t cataLTree

-- mirror

g1 = Leaf
:t g1

g2 = Fork . swap
:t g2

mirror = cataLTree (either g1 g2)
:t mirror

-- testing

mirror (Fork (Fork (Leaf 1,Leaf (2)),Leaf (3)))
```

```
Leaf :: forall a. a -> LTree a

Fork :: forall a. (LTree a, LTree a) -> LTree a

outLTree :: forall a. LTree a -> Either a (LTree a, LTree a)

cataLTree :: forall b d. (Either b (d, d) -> d) -> LTree b -> d

g1 :: forall a. a -> LTree a

g2 :: forall a. (LTree a, LTree a) -> LTree a

mirror :: forall a. LTree a -> LTree a

Fork (Leaf 3,Fork (Leaf 2,Leaf 1))
```