2. O combinador

$$\mathsf{const} :: a \to b \to a$$
$$\mathsf{const}\ a\ b = a$$

está disponível em Haskell para construir funções constantes, sendo habitual designarmos const k por $\underline{k}$, qualquer que seja k. Demonstre a igualdade

$$\underline{(b,a)} = \langle \underline{b}, \underline{a} \rangle \tag{F1}$$

a partir da propriedade universal do produto e das propriedades das funções constantes que constam do formulário.

Resolução

$$\underline{(b,a)} \quad = \quad < \underline{b}, \underline{a} >$$

{ propriedade universal do produto, **lei (6)**, fazendo $k = \underline{(b,a)}$ }

$$\begin{aligned} \pi_1 \cdot \underline{(b,a)} &= \underline{b} \\ \pi_2 \cdot \underline{(b,a)} &= \underline{a} \end{aligned}$$

{ fusão-const, **lei (4)** }

$$\begin{aligned} \underline{\pi_1 (b,a)} &= \underline{b} \\ \underline{\pi_2 (b,a)} &= \underline{a} \end{aligned}$$

{ def-proj, **lei (79)** }

$$\begin{aligned} \underline{b} &= \underline{b} \\ \underline{a} &= \underline{a} \end{aligned}$$

{ propriedade reflexiva da igualdade }

$$\mathit{True}$$
$$\mathit{True}$$

In [1]:

```
-- testing with const (b,a) = < const 2, const "a" >

split f g x = (f x, g x)

:t const (2,"a")
:t split (const 2) (const "a")
```

```
const (2,"a") :: forall a b. Num a => b -> (a, [Char])
split (const 2) (const "a") :: forall a b. Num a => b -> (a, [Char])
```