Cálculo de Programas

2.° ano
Lic. Ciências da Computação e Mestrado Integrado em Engenharia Informática
UNIVERSIDADE DO MINHO

2020/21 - Ficha nr.° 3

1. Considere o isomorfismo

$$(A+B)+C \underset{\text{coassocl}}{\overset{\text{coassocr}}{\cong}} A+(B+C)$$

onde coassocr $= [id+i_1, i_2 \cdot i_2]$. Calcule a sua conversa resolvendo em ordem a coassocl a equação,

$$\text{coassocl} \cdot \text{coassocr} = id$$

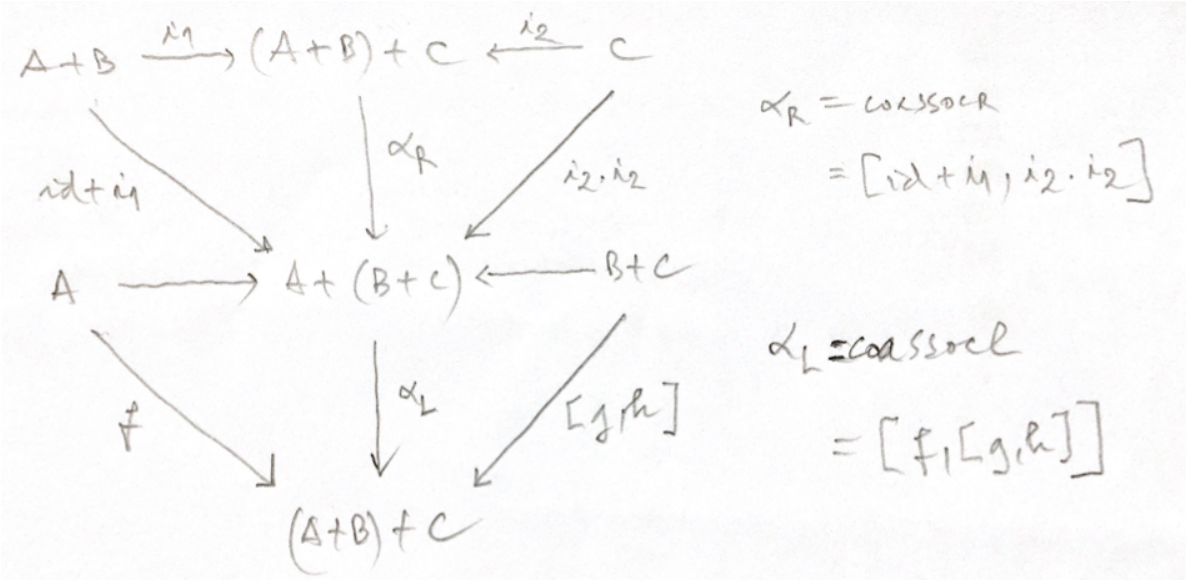
isto é

$$\text{coassocl} \cdot \underbrace{[id+i_1, i_2 \cdot i_2]}_{\text{coassocr}} = id$$

etc. Finalmente, exprima coassocl sob a forma de um programa em Haskell *não recorra* ao combinador "either".

Resolução

Vamos começar por desenhar o diagrama que testemunha o isomorfismo.



Ou seja, verificamos que $coassocl$ corresponde a uma função do tipo $[f, [g, h]]$, para:

- $A \xrightarrow{f} (A+B)+C$
- $B+C \xrightarrow{[g,h]} (A+B)+C$

Calculemos então $[\, f, [g, h]\,] \cdot coassocr = id$ em ordem a f, g e h.

$[f, [g, h]] \cdot [id + i_1, i_2 \cdot i_2] = id$

{ fusão-+, **lei (20)**; reflexão-+, **lei (19)** }

$$\equiv [\,[f,[g,h]]\cdot(id+i_1)\;,\;[f,[g,h]]\cdot(i_2\cdot i_2)]\;=[i_1,i_2]$$

{ absorção-+, **lei (22)** }

$$\equiv [\,[f\cdot id,[g,h]\cdot i_1]\;,\;[f,[g,h]]\cdot(i_2\cdot i_2)]\;=[i_1,i_2]$$

{ cancelamento-+, **lei (18)**; natural-id, **lei (1)**; assoc-comp, **lei (2)** }

$$\equiv [\,[f,g]\;,\;([f,[g,h]]\cdot i_2)\cdot i_2]\;=[i_1,i_2]$$

{ duas vezes cancelamento-+, **lei (18)** }

$$\equiv [\,[f,g]\;,h]\;=[i_1,i_2]$$

{ eq-+, **lei (27)** }

$$\equiv [f,g]=i_1\;;\;h=i_2$$

{ reflexão-+, **lei (19)** }

$$\equiv [f,g]=i_1\cdot[i_1,i_2]\;;\;h=i_2$$

{ fusão-+, **lei (20)** }

$$\equiv [f,g]=[i_1\cdot i_1,i_1\cdot i_2]\;;\;h=i_2$$

{ eq-+, **lei (27)** }

$$\equiv f=i_1\cdot i_1\;;\;g=i_1\cdot i_2\;;\;h=i_2$$

Ou seja,

$$coassocl=[f,[g,h]]=[i_1\cdot i_1,[i_1\cdot i_2,i_2]]$$

{ natural-id, **lei (1)** }

$$=[i_1\cdot i_1,[i_1\cdot i_2,i_2\cdot id]]$$

{ def-+, **lei (21)** }

$$=[i_1\cdot i_1,i_2+id]$$

Temos assim finalmente

$$coassocl=[\,i_1\cdot i_1,i_2+id]$$

Haskell

In [1]:

```
:load ../src/Cp.hs

-- pointfree

coassoclPF = either (i1 . i1) (i2 -|- id)

-- pointwise
```

```
coassoclPW (Left a) = Left . Left $ a
coassoclPW (Right (Left a)) = Left . Right $ a
coassoclPW (Right (Right a)) = Right a
```

In [2]:

```
-- type checking

:t coassoclPF
:t coassoclPW
```

```
coassoclPF :: forall a b1 b2. Either a (Either b1 b2) -> Either
(Either a b1) b2
```

```
coassoclPW :: forall a b1 b2. Either a (Either b1 b2) -> Either
(Either a b1) b2
```