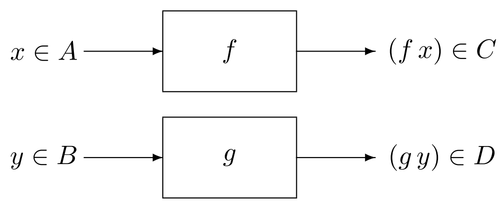
Cálculo de Programas

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

2018/19 - Ficha nr.° 2

1. O diagrama de blocos



descreve o combinador funcional *produto*

$$f \times g = \langle f \cdot \pi_1, g \cdot \pi_2 \rangle \tag{F1}$$

que capta a aplicação *paralela* e *independente* de duas funções $A \xrightarrow{f} C$ e $B \xrightarrow{g} D$:

$$\begin{tikzcd} A \arrow[d, "f"'] & B \arrow[d, "g"'] & A \times B \arrow[d, "f \times g"] \\ C & D & C \times D \end{tikzcd}$$

(a) Mostre que $(f \times g)\ (x, y) = (f\ x, g\ y)$.

(b) Mostre ainda que

$$\pi_1 \cdot (f \times g) = f \cdot \pi_1$$
$$\pi_2 \cdot (f \times g) = g \cdot \pi_2$$
$$id \times id = id$$
$$(f \times g) \cdot (h \times k) = f \cdot h \times g \cdot k$$

Desenhe os diagramas destas igualdades.

2. Preencha da forma mais genérica possível os "?" do diagrama:

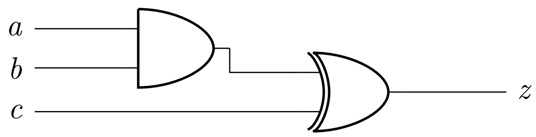
$$\begin{tikzcd} ? & ? \arrow[l, "\langle \pi_2, \pi_1 \rangle"'] & ? \arrow[l, "\langle \pi_2, \pi_1 \rangle"'] \arrow[ll, bend left, "id"] \end{tikzcd}$$

3. Considere as funções seguintes:

$$f = \langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle$$
$$g = \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle$$

Identifique os tipos de f e g. Acompanhe a sua resolução com a construção dos respectivos diagramas.

4. Considere o circuito booleano



que calcula a função $f\ ((a, b), c) = (a \wedge b) \oplus c$, onde $\oplus$ é a operação "exclusive-or".

- Escreva uma definição dessa função $(\mathbb{B} \times \mathbb{B}) \times \mathbb{B} \xrightarrow{f} \mathbb{B}$ que não recorra às variáveis a, b ou c[1] e desenhe o respectivo diagrama.
- Qual é o tipo da função $g = \langle \pi_1, f \rangle$?

5. Suponha que tem a relação $db_1 \in (Dat \times Jog^*)^*$ de todos os jogos que se efectuaram numa dada competição, organizados por data. Seja ainda $db_2 \in (Jog \times Atl^*)^*$ a indicação, para cada jogo, dos atletas que nele participaram.

Um comentador desportivo pede-lhe que derive de db_1 e de db_2 a relação, ordenada por nome, das datas em que cada atleta jogou, datas essas também ordenadas:

$$f : (Dat \times Jog^*)^* \to (Jog \times Atl^*)^* \to (Atl \times Dat^*)^*$$
$$f\ db_1\ db_2 = \ldots$$

Mostre que f pode ser escrita numa só linha usando os combinadores $f \cdot g$, $f \times g$, etc que até agora estudou, desde que tenha à sua disposição a seguinte biblioteca de funções **genéricas**:

- $sort : A^* \to A^*$
  — ordena listas de A segundo uma ordem previamente assumida (sobre A)
- $collect : (A \times B)^* \to (A \times B^*)^*$
  — agrupa uma sequência de pares segundo os respectivos primeiros elementos, e.g.
  $collect\ [(1, 2), (5, 6), (1, 3)] = [(1, [2, 3]), (5, [6])]$
- $discollect : (A \times B^*)^* \to (A \times B)^*$
  — inversa da anterior
- $converse : (A \times B)^* \to (B \times A)^*$
  — troca os elementos de cada par entre si
- $comp : (A \times B)^* \to (B \times C)^* \to (A \times C)^*$
  — encadeia as sequências de entrada de acordo com os elementos em comum (de tipo B).

[1] Definições de funções que recorrem a variáveis dizem-se "*pointwise*"; as correspondentes versões sem variáveis dizem-se "*point-free*".