What associativity is about

We all know what it means for a binary operator Δ to be associative:

$$(a \Delta b) \Delta c \;=\; a \Delta (b \Delta c) \quad .$$

Now, if you interview people what associativity is really about, you get as one of the first answers that you are allowed to omit the parentheses, and write

$$a \Delta b \Delta c$$

without introducing ambiguity. And that is right!

Next, you may get the answer that thanks to the associativity it does not matter whether the value of

$$a \Delta b \Delta c$$

is computed from left to right or from right to left.

And you may get several other answers, but never the one to be explained next (at least outside my own circles).

* * *

For people that calculate, the property of associativity may offer very strong heuristic guidance: If, in a calculation, the expression

$$a \Delta b \Delta c$$

enters the picture as

$$(a \triangle b) \triangle c \quad ,$$

then the advice for the next step(s) is to focus on the subexpression $b \triangle c$ in

$$a \triangle (b \triangle c) \quad .$$

In our experience, this heuristic rule has worked well in a tremendous amount of cases. And if you come to think of it, of course: what else could associativity be about?

* * *

The proof of the pudding is in the eating, and we now shall present the proof of a FANTASTIC theorem that Paul F. Hoogendijk challenged me to prove today. The theorem is FANTASTIC to the extent that I have absolutely no idea what the mathematics involved "means".

In what follows constant Γ and dummies x and y are of type Thing. Further, $\wedge$ and E map Things to Things, and last but not least, binary operator $\circ$, which maps two Things on a Thing, is associative. In short, in the ensuing expressions there is no type conflict.

Now, given (0) and (1)

(0) $\quad x = \Lambda.y \quad \equiv \quad \Gamma \circ x = y \qquad (\forall x,y)$

(1) $\quad E.x = \Lambda.(x \circ \Gamma) \qquad (\forall x)$

we have to prove

(2) $\quad E.x \circ \Lambda.y = \Lambda.(x \circ y) \qquad (\forall x,y)$.

<u>Proof</u> (0) is a so-called Galois-connection, and the advice is that we then always immediately write down the corresponding cancellation rules. Here, they are

(3) $\quad \Gamma \circ \Lambda.y = y \qquad (\forall y)$

(4) $\quad \Lambda.(\Gamma \circ x) = x \qquad (\forall x)$

Now, we tackle (2) and, for this occasion, we shall explicitly devote a step that appeals to the associativity of $\circ$.

$\quad E.x \circ \Lambda.y = \Lambda.(x \circ y)$

$\equiv \qquad$ { (0) from left to right }

$\quad \Gamma \circ (E.x \circ \Lambda.y) = x \circ y$

$\equiv \qquad$ { $\circ$ is associative }

$\quad (\Gamma \circ E.x) \circ \Lambda.y = x \circ y$

$\equiv \qquad$ { (1), to eliminate E }

$\quad (\Gamma \circ \Lambda.(x \circ \Gamma)) \circ \Lambda.y = x \circ y$

$\equiv \qquad$ { (3) with $y := x \circ \Gamma$ }

$\quad (x \circ \Gamma) \circ \Lambda.y = x \circ y$

$$\equiv \quad \{ \circ \text{ is associative} \}$$

$$x \circ (\Gamma \circ \Lambda . y) = x \circ y$$

$$\equiv \quad \{ (3) \}$$

$$x \circ y = x \circ y$$

$$\equiv \quad \{ \}$$

true .

(<u>End</u> of Proof.)

"Ik wist niet dat ik het in me had"

M. Toonder

"I never knew I was that smart".

W.H.J. Feijen

Sterksel,
22 September 1994