Direct Functions in Dyalog APL

John Scholes – Dyalog Ltd.
john@dyalog.com

A Direct Function (dfn) is a new function definition style, which bridges the gap between named function expressions such as `rank←⍴∘⍴` and APL's traditional 'header' style definition.

Simple Expressions

The simplest form of dfn is: **{expr}** where **expr** is an APL expression containing `'⍺'`s and `'⍵'`s representing the left and right argument of the function respectively. For example:

```
      {(+/⍵)÷⍴⍵} 1 2 3 4          ⍝ Arithmetic mean
2.5
      2 {(⍺-⎕io)+⍳1+⍵-⍺} 5        ⍝ Sequence ⍺ ... ⍵
2 3 4 5
      {{⍵*0.5}+/⍵*2} 3 4          ⍝ Pythagorus (Note inner D-Fn).
5
```

A dfn can be used in any function context ...

```
      {⍺+÷⍵}/25⍴1                 ⍝ Repeated fraction.
1.618033989
```

... and of course, assigned a name:

```
      root←{⍵*÷⍺}                 ⍝ ⍺'th root.
```

Dfns are ambivalent. Their right (and if present, left) arguments are evaluated irrespective of whether these are subsequently referenced within the function body.

```
      2{⍵}3
3
      4{⍺}5
4
```

Guards

A guard is a boolean-single valued expression followed by `':'`. A simple expression can be preceded by a guard, and any number of guarded expressions can occur separated by `'⋄'`s.

```
      ⍺<0:                       ⍝ Left arg negative.

      0=≡⍵:                      ⍝ Right arg simple scalar.
```

Guards are evaluated in turn (left to right) until one of them yields a 1. Its corresponding **expr** is then evaluated as the result of the dfn. A guard is equivalent to an If-Then-Else or Switch-Case construct.

```
      {⍵=0:'zero' ⋄ ⍵≠0:'non zero'} 3
non zero
```