

J を品詞により理解する —J Quick Reference (ver.2) を傍らに—

西川 利男

J においては、プリミティブ (原始語)、ユーザ定義語を問わず、品詞という属性があり、これによりそれぞれの語の働きがきまる。

品詞の種類

aa という定義語の品詞は、4!:0 @ < aa により返される値で調べられる。

0 名詞 1 副詞 2 接続詞 3 動詞 6 ロケール

名詞

名詞 DA のデータ型は、3!:0 DA により示される。

1 boolean 2 literal 4 integer 8 floating 16 complex
32 boxed 64 extended integer 128 rational

名詞の大きさ、形には、次のものがある。

スカラー (= atom), ベクトル (= list), 配列 (2次元、3次元)

動詞

動詞のはたらく順序は原則として「右から左へ」である。… 関数型言語
引数をとる数により、単項動詞と2項動詞とがある。

名詞と動詞とのかかわり合いを決めるものにランクがある。たとえば動詞*に対して

* b. 0 => _ 0 0

のようにランクが示され、これは、単項の引数としては、ランク_の名詞 (どんなランクでもよい) をとり、2項の引数としては、ランク0の名詞をとることを意味する。

明示定義(Explicit)

引数をあらわに示して、3 : または 4 : (2項動詞のときのみ)により、実行順序のとおり記述する。

次のようないろいろな制御構文が使える。

while. do. end.

if. do. else. end.

暗黙定義(Tacit)

引数を示さない特別な構文 (fork や hook を用いた) により、極めてコンパクトな定義ができる。

副詞

引数を1つ (動詞でも良い) とる。動詞と組み合わせて、新しい動詞をつくる。

たとえば、副詞(/)は動詞(+)と組み合わせて、合計(+/)を求める動詞が定義できる。

接続詞

引数を2つ (動詞でも良い) とる。これも動詞と組み合わせて、新しい動詞をつくる。
たとえば、接続詞(^: power)は、繰り返しをおこなうので、2倍する動詞(+:)と組み合わせて、+:^(2) 3 => 12 (= 3 x 2 x 2) のようになる。

NB. Examples of Explicit Definition =====

```
mean =: 3 : 0
(+/ y) % (# y)
)
```

NB. mean 1 2 3 4 5 4 3 2 1 => 2.77778

```
stdev =: 3 : 0
m =. (+/ y) % (# y)
ss =. +/ *: (y - m)
%: (ss % (# y))
)
```

NB. stdev 1 2 3 4 5 4 3 2 1 => 1.31468

```
primesto =: 3 : 0
IN =. >: i. y
(2 = +/ 0 = IN | / IN) # IN
)
```

NB. primesto 50

NB. 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

NB. Examples of Tacit Definition =====

```
mean0 =: +/ % #
stdev0 =: %:@(mean0 @(*: @[] - mean0) ) )
primes0 =: (2: = (+/@(0: = i. | / i.) ) ) # i.
```

```
mean 1 2 3 4 5 4 3 2 1
2.77778
```

```
mean0 1 2 3 4 5 4 3 2 1
2.77778
```

```
stdev 1 2 3 4 5 4 3 2 1
1.31468
```

```
stdev0 1 2 3 4 5 4 3 2 1
1.31468
```

```
primes0 20
2 3 5 7 11 13 17 19
```

NB. Primes Idiom =====

```
primes_to =: i.&. (p:^:_1)
primes_tol =: p:@i.@(p:^:(_1))
```

```
primes_to 20
2 3 5 7 11 13 17 19
```

```
primes_tol 20
2 3 5 7 11 13 17 19
```