

A propos de l'Epsilon souligné par Alain Delmotte

Dans le numéro 20 des Nouvelles, Gérard Langlet demandait quel est l'équivalent J du « Find » (Ⓢ epsilon souligné) d'APL*PLUS.

Dans le numéro 21 j'avais proposé une fonction explicite. Mais dans le même numéro, Michel Dumontier en page 98 écrit :

« Réponse : c'est E. voir les Nouvelles d'APL N° 16 de septembre 1995, page 64, Cours de J leçon 3, j'avais pourtant donné un exemple qui aurait dû frapper les esprits ! »

Je le reproduis ici :

```
? 'co' E. 'cocoricoco-co'  
1 0 1 0 0 0 1 0 1 0 0 1 0
```

NB. il faut lire les nouvelles d'APL avec un peu plus d'attention !... »

..... !!

Après avoir refait les tests sur les exemples de STSC/APL*PLUS et constaté que E. ne répond pas entièrement à la définition de epsilon souligné, j'ai adressé un message au « newsgroup » comp.lang.apl demandant comment écrire, sous forme tacite, les fonctions proposées dans le numéro 21 des Nouvelles.

Voici la traduction de la réponse de Roger Hui (elle commence par une reprise de mon message) :

Alain Delmotte écrit le dimanche 5 janvier :

```
> Dans le numéro 20 des « Nouvelles d'APL », la revue de  
> l'Association francophone d'APL, Gérard Langlet demandait  
> quel est l'équivalent J de la fonction « Find » (epsilon  
> souligné) d'APL*PLUS.  
>  
> Dans le numéro 21, Michel Dumontier répond que c'est E.  
> (membre d'intervalle), mais j'ai proposé, dans le même  
> numéro, les fonctions suivantes qui utilisent la forme  
> diadique du verbe « couper » avec un coefficient 3 (tuiles) :  
>  
> eun=: 3 : 0  
> :  
> c=.1, ,: $x.      NB. taux de glissement des sous unités de y.  
>                  NB. à comparer à x. ; le taux de glissement  
>                  NB. est 1 dans chaque direction  
> d=. $ $ x.      NB. rang de la comparaison  
> x.-:"d c{;.3 y.  
> )  
>  
> eu=: 3 : 0  
> :
```

```

> x. -:"($$x.) (1,,:$x.) [;.3 y.
> )
>
> a=. 'co'
> b=. 'cocoa'
>
> et pour l'exemple du « Quick Reference Guide » d'APL*PLUS
> (Ch.H. Lee, pour le PC, STSC, 1986)
> c=. 2 2 $ 2 3 3 1
> d=. 3 4 $ 1 2 3
>
> a E. b
> 1 0 1 0 0
> c E. d
> |nonce error
> | c E.d
> a eun b
> 1 0 1 0 0
> c eun d
> 0 1 0 0
> 1 0 0 0
> 0 0 0 0
>
> Est-il possible d'obtenir une réponse correcte avec E. et un
> rang approprié ?
> Comment écrire eun (eu) sous forme tacite ?

```

E. n'est implémenté pour le moment que pour des arguments chaînes (vecteurs). D'où le « nonce error ».

Une implémentation de E. en terme de tuiles peut en effet être écrite sous forme tacite. A partir de "eu" ci-dessus,

```

eu=: 3 : 0
:
x. -:"($$x.) (1,,:$x.) [;.3 y.
)

```

les simplifications suivantes sont possibles :

- a. Une définition explicite exclusivement diadique peut être écrite plus succinctement comme 4 : blabla
- b. 1,,:vec est équivalent à 1,:vec
- c. un argument gauche 1,:vec pour les tuiles est équivalent à un argument gauche de vec seul.

Donc :

```

eu1= : 4 : 'x. -:"($$x.) ($x.) [ ;.3 y.'

```

De plus, ici [;.3 est utilisé pour générer un tableau de cellules de taille \$x. , et chaque cellule est comparée à x. Ceci peut être fait de deux autres manières différentes : la génération de cellules et les comparaisons peuvent être combinées en une seule étape, donc :

```
eu2= : 4 : '($x.) (x.&-:);.3 y.'
```

De manière alternative, < ;.3 peut être utilisé à la place de [;.3, et la comparaison qui la suit peut donc être (<x.)=blabla , donc :

```
eu3= : 4 : '(<x.) = ($x.)<;.3 y.'
```

Cette dernière définition peut être écrite tacitement, simplement :

```
13 : '(<x.) = ($x.)< ;.3 y.'  
([: < [] = ([: $ [] < ;.3 ]
```

Je l'aurais personnellement écrite de la façon suivante :

```
eu4= : <@[ = $@[<;.3 ]  
-----
```

Ceci montre, si besoin était, la très grande clarté d'analyse et de conception de Roger Hui, moins parce qu'il a écrit l'interprète de J que par sa parfaite connaissance de l'essence même du langage.

D'autre part Roger Hui donne un « Cours de J », clair et concis, sur les étapes à suivre pour l'amélioration d'une fonction de sa forme explicite vers une forme tacite.