Subject: Re: Efficient pattern-matching in a large array
Posted by JDS on Thu, 18 Aug 2011 17:00:14 GMT
View Forum Message <> Reply to Message

Right you are, Bob! Thanks for the catch.

I struggled to find a simple logical test with the properties:

1) any array is true
2) !NULL is false

And I didn't remember that KEYWORD_SET looks only at the first entry. I think *any array* should be true on its own, but IDL logic treats scalars and single length arrays as identical, and issues an error when IF-testing longer arrays. This is maximally confusing, since you often don't know the tested array's length in advance, and an error is not thrown when that length drops to 1. I think David has an article somewhere about the resulting confusion.

In any case, you don't have to give up on !NULL for the test. In fact you can take !NULL even further, applying it without testing for WHERE's success at all (one of the main uses of !NULL). This further simplifies the algorithm:

```
w=where(a eq pat[(c=0)],/NULL)
while w ne !NULL && ++c lt n_elements(pat) do w=w[where(a[w+c] eq pat[c],/NULL)]
w=w[where(w le n-n_elements(pat),/NULL)]
```

This will work even if the pattern occurs right away. That last test is new and is for one corner case: if the pattern contains a string of repeating values at its end, and this value matches the very last element of the array, you will otherwise get a spurious match.

By the way, this uses the curious fact that comparison to !NULL is special cased in IDL (since you can't have an array of !NULL's):

```
IDL> a=[1,2,4,5]
IDL> print, a ne 2
   1   0   1   1
IDL> print, a ne !NULL
   1
```

JD