Subject: Re: idl2matlab translate-o-matic
Posted by John-David T. Smith on Thu, 24 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

```
>
>  "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>>>  Since I can't pass a testing function to that routine (IDL doesn't have
>>>  higher order functions), I will accept a routine, for illustrative purposes,
>>>  that removes all even values from the array.
>>>
>>>  Now suppose some joker passes an array containing only even values to that
>>>  routine...
>>>
>>>  - DM
>>>
>>
>>  wh=where(array mod 2, cnt)
>>  if cnt gt 0 then return,array[wh] else return, -1
>>
>>  I use scalars (often -1) as cheap and easy to use empty arrays.  Anything with:
>>
>>   size(x,/N_DIMEN) eq 0
>>
>>  is patently *not* an array.
>>
>>
>>  And as far as the lack of "higher order testing functions":
>>
>>  function evens, arr
>>     return, arr mod 2 eq 0
>>  end
>>
>>  function odds, arr
>>     return, arr mod 2
>>  end
>>
>>  function exclude,arr, exc_func
>>     wh=where(call_function(exc_func,arr) eq 0,cnt)
>>     if cnt gt 0 then return,arr[wh] else return,-1
>>  end
>>
>>  and to get rid of the odds, e.g.:
>>
>>  IDL> a=exclude(b,"odds")
>
>  Okay, but let's say now you wanted to merge two lists like that
>  together.  Wouldn't this be nice:
```

>
> IDL> c = [exclude(a,'odds'), exclude(b,'evens')]
>
> The way I say it makes it sound like it's just an inconvenience, which
> it is. But for gosh sakes, its a *completeness* issue too. We don't
> have a general purpose number system without zero! It would be silly.
> Why should we have lists without the empty list? Instead we have to
> drag around this extra notion of the COUNT or play tricks by returning
> scalars.
>

I totally agree with you about the convenience of such an entity. I'm not trying to deny that. What I'm trying to show is that what we might think of as an "empty array" or "empty list" is just an abstract notion, actually implemented in code in some way analogous to what I've done. In stark contrast to the issue you raise of a complete number system, in which the internal representation for "0" is equivalent to that for any other number, an empty array is achieved only by special case programming, which just happens to be hidden from our sight. Now, IDL is not C, and lots of special case programming is hidden from our sight, so I'm certainly not arguing that hidden conveniences, if well implemented, are to be avoided. I just want everyone to understand that this would be an addition purely motivated by convenience, and that there really is no fundamental "incompleteness".

Having said that, I see no reason that it couldn't be done pretty easily. Variables can already be marked "undefined", so why not extend that somewhat and allow "undefined" arrays and lists to exist. Dimensionality is important of course, so the concept of a 2x2 empty array need be addressed, etc., but I wouldn't think it's prohibitive.

JD

--
J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|