Subject: Re: Matching 2 lists
Posted by David Baker on Sat, 21 Aug 2010 17:03:39 GMT
View Forum Message <> Reply to Message

On Aug 21, 5:52 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Aug 21, 1:37 pm, David Baker <de...@le.ac.uk> wrote:
>
>>>> If you're using IDL 8.0, you could modify match_2d to return a list of
>>>> length n_a, where each element is an array of indices into b.
>
>>> Thanks for the advice, whilst I maybe comfortable programming basic
>>> stuff I can't even begin to follow the match_2d code so wouldn't know
>>> where exactly to code in what you suggest. But that is exactly what
>>> I'm after.
>
>>> -Cheers,
>>> David
>
>> You mentioned using IDL 8.0. My university is currently still on
>> 7.1.1, is there anything unique about IDL 8.0 to the solution you
>> propose?
>
> Lists and empty arrays. In that case, because each element of A can
> have a different number of matches, so each element of the list is an
> array of a different size (possibly empty, if there are no matches).
>
> The most direct way to do something similar in IDL 7 is a pointer
> array, where each element points to the array of indices that match
> the corresponding element of A. Which is more awkward to use due to
> the need to dereference the pointer, and the need to test (with null
> pointers, for instance) for the no-match case.

OK, but it should still be possible? You're suggesting something like
a two element structure, one element contains the index from A the
other element contains a pointer listing the indicies of any object
from list B that is within the search radius? The structure would just
need to grow via a=[b,c] for each object in A that has matches to
those in B (though this would obviously cause a speed penalty I think
it might outweigh that caused by looping over my lists). The tricky
part I guess (at least for me) is finding out where match_2d prints
out the indicies from B that match to each individual index in A.

-David