Subject: Re: Regular Expressions
Posted by Wayne Landsman on Fri, 16 Mar 2001 03:24:50 GMT
View Forum Message <> Reply to Message

JD Smith wrote:

> IDL> print, stregex(st,'(^|[^l])l($|[^l])')
>
> which means "a character that is not 'l', or the beginning of the
> string, followed by an 'l', followed by a character that is not 'l', or
> the end of the string". Aren't you glad Ken Thompson didn't decide
> originally to develop regexps in english?
>
> This will also work on
>
> IDL> st = "let's all go the the movies"

Thanks. But I now realize that my original formulation was not quite correct, since the above expression (usually!) returns the position of the character *before* the 'l', so to get the position of the first single 'l' one has to add 1

IDL> l_position = stregex(st,'(^|[^l])l($|[^l])') + 1

Unfortunately, if 'l' is the first character, then you *don't* want to add the 1. (The expression stregex(st,'(^|[^l])l($|[^l])') returns a value of 0 for both st ='long days' and st ='slow nights'.)
One solution is to forget about the beginning of string anchor and just concatenate a blank to the beginning to the string

IDL> l_position = stregex(' ' + st,'[^l]l($|[^l])')

--Wayne

P.S. The real-life problem I am working on deals not with 'l' but with apostrophes. I am trying to speed up the processing of FITS header values, where is a string is delineated by non-repeating apostrophes, and a possessive is indicated by a double apostrophe.

VALUE = 'This is Wayne''s FITS value' / Example string field