WORDFNS

By: Ron Kaplan (Kaplan.pa@Xerox.com)
Becky Burwell (Burwell.pa@Xerox.com)

Uses: SETSTRINGLENGTH

This document last edited on August 19, 1988.

INTRODUCTION

WORDFNS is a set of functions for manipulating files of words. There are functions to do the following: sort files, manipulate sorted files, provide common i/o functions for word files, provide mapping and translation mechanisms, provide common translation functions, and provide packaged mapping utilities.

The idea behind the mapping mechanism is that you can translate a file or list of files by specifying a read function to operate on each chunk of a file (the obvious two chunks are words and lines). You can specify file specific translation functions, default functions (when file specific functions are not provided) and common translation functions for all files. The input to the first translation function is the result of applying the read function to an input stream open on a file. The output of the first translation function is passed as input to the second translation function, etc.

USE

Note: for any file, if NIL or T is specified then the results are printed in the executive window.

Sorting Files

(SORTWORDFILE *IFILES OFILE COMMONTRANSFNS DEFAULTTRANSFNS READFN COMMONCOMPAREFN KEEPDUPLICATES FIELDS SEPARATOR REVERSEORDERFLG FASTFLG*) [Function]

The functions sorts the words on IFILES and stores the result back on OFILE. Th duplicates are eliminated unless *KEEPDUPLICATES* is non-NIL. For a description of the function of the argumentsCOMMONTRANSFNS, DEFAULTTRANSFNS and READFN see the section entitled "Translation Mechanisms". The argument *FIELDS* is used to specify the sorting order. The separator of the fields is specified in *SEPARATOR*. If *REVERSEORDERFLG* is T the result of the sort is reversed. *FASTFLG* set T causes the sort to caches the fields by consing allowing for a quicker sort (but consumes memory).

FIELDS is one of: NIL, a list of field numbers or else a list of one, two or three element lists of the form: FieldNumber Type CompareFn where type is either STRING (the default) or NUMBER. The default comparefn for STRING is ALPHORDER; for NUMBER is NUMORDER [Argument]

SEPARATOR is one of the following: a character string, a bittable, a list of single character atoms or numbers or one of the special atoms WHITESPACE (indicating a space or tab) or the atom TAB. The default is WHITESPACE.

[Argument]

>> should I put NUMORDER and GetNthField here?<<

Note: two related functions, NUMORDER and GetNthField, are described in the miscellaneous section.

Functions for use with sorted files

In each of the following functions:

COMMENTFILE contains the details of the result of the function (for example, the number of strings that were read in from each file) [Argument]

(COMMONSORTEDFILES *file1 file2 ofile COMMENTFILE*) [Function]

Computes the intersection of two sorted files, *file1* and *file2* and the results are stored on *ofile*. The files are read a line at a time. The value is the full name of *ofile*.

(COMPARESORTEDFILES *file1 file2 ofile IMINUS2 2MINUS1*

COMMENTFILE COMMENT) [Function]

The two sorted files, *file1* and *file2*, are compared a line at a time. The common lines are stored on *ofile*. The output is in two colums: the left column for those lines in *file1* that do not exist in *file2* and the right column for those lines in *file2* that do not exist in *file1*. The two flags *IMINUS2* and *2MINUS1* are used to determine how the the comparisons will be performed. If they are not specified they are both assumed to be T thus meaning that the comparison will be performed by subtracting *file2* from file *file1* and *file2* with *file1* substracted. If only one of *IMINUS2* or *2MINUS1* is specified then only the specified one way comparison will be done. *COMMENT* is intended to be a string which, by default, is the string "Comparison". This string is inserted at the top of the file. The value is the full name of *ofile*.

(DIFFSORTEDFILES *FILE1 FILE2 OUTFILE COMMENTFILE*) [Function]

The result of subtracting *FILE1* from *FILE2* is stored on *OUTFILE*. The files are read a line at a time. The value is the full name of *ofile*.

I/O Functions

In the two major read functions, DREADLINE and DREADWORD, the SPACE, CR, LF and ^Z in any character set are interpreted to be the corresponding character in character set zero.

(DREADLINE *stream string skipsemicolons*) [Function]

Words are read from the word-stream *stream*, smashing them into *string*, which grows as needed. Returns NIL at EOF. Skips leading and trailing separators, and if *skipsemicolons* is non-NIL then sequences from ";" to EOL are treated as a composite separator or end-marker. Unlike DREADWORD (desribed later in this section), segments are separated only by EOL, so compounds are not split into components. Note that *stream* must have been set up so that BIN/READCCODE returns NULL on EOF.

(DREADLINESKIPSC *stream string skipsemicolons*) [Function]

Calls DREADLINE with *skipsemicolons* bound to T.

(DREADWORD *stream string*) [Function]

Words are read from the word-stream *stream*, smashing them into *string*, which grows as needed. Returns NIL at EOF. Skips leading and trailing separators, and treats sequences from ";" to EOL as a composite separator or end-marker. Unlike DREADLINE, segments are separated by space as well as EOL, so splits compounds into components. Note that *stream* must have been set up so that BIN/READCCODE returns NULL on EOF.

(INPUTWORDSTREAM *FILE NOPRINT*) [Function]

Returns a stream that is guaranteed to be open for word-reading (e.g. using DREADLINE or DREADWORD) at the beginning of *FILE*. If *NOPRINT* is NIL then the fullname of the file will be output.

(OUTPUTWORDSTREAM *FILE*) [Function]

Returns and opens a stream for the output of words (sequential text) guaranteed closed when reset context is exited and deleted if there is an error.

Translation Mechanisms

Translation mechanisms are supplied to allow great flexibility in translating one or more files which may be in different formats and have unique translations applied to them. To specify how a file is to be read a read function (READFN) can be specified. Two common read functions, described previously, are DREADLINE and DREADWORD.

As mentioned earlier each file can have unique or common translation functions. A translation function is a function which takes two arguments: a string (the input to be translated) and an optional scratch string which can be destructively modified. The output of the translation function is one of the following: a string, the value T or the value NIL. Readers may wish to refer to the LispUsers module SETSTRINGLENGTH. The special value T denotes that the output is the same as the input. A value of NIL means that nothing will be kept.

>> a better name for translation set <<

>> what is the syntax?<<

IFILES is either a single file name, a single translation set, a list of file names or a list of translation sets. Translation sets have the form: (READFN *READFN TRANFN1 TRANSFN2* ...). (Note the first element of the list is the actual atom READFN and the second element [Argument]

READFN is the read function that is used for reading the files. It is passed a stream. Unless specified otherwise, the default read function is DREADWORD. [Argument]

DEFAULTTRANSFNS is a single function or list of functions which is first applied to the first file. What is given to the translation function is determined by what the read function passed. Unless specified otherwise, the default read function is DREADWORD. The result of applying the first function in DEFAULTTRANSFNS is input to the second function in DEFAULTTRANSFNS. This result is then passed on for application to the functions in COMMONTRANSFNS. The special value T denotes that the output is the same as the input. [Argument]

COMMONTRANSFNS [Argument]

DONTPRINT is the argument which decides whether the details of the translation functions will be printed. By default it is NIL meaning that the details will be printed. [Argument]

(TRANSLATEWORDFILE *IFILES COMMONTRANSFNS DEFAULTTRANSFNS READFN DONTPRINT*) [Function]

TRANSLATEWORDFILE produces an output file by translating each word in (possibly a list of) *IFILES* through a translation function. List elements of files are paired with their own idiosyncratic translation function. Otherwise the *DEFAULTTRANSFNS* is used. *COMMONTRANSFNS* are applied to the results of the default or file-specific translations to produce the translation string. If any translation function returns NIL, that string is skipped. A translation function is assumed to be an identity if it returns T, which makes simple predicates easy.

(COLLECTWORDFILE *IFILES COMMONTRANSFNS DEFAULTTRANSFNS READFN DONTPRINT*) [Function]

Returns the list of non-NIL values of functions applied to words in *IFILES*.

(MAPWORDFILE *IFILES COMMONTRANSFNS DEFAULTTRANSFNS MAPFN READFN DONTPRINT*) [Function]

Maps mapping function over words in *IFILES*. Nothing is setup for output.

Packaged Mapping Utilities

(LONGESTWORDS *FILES COMMONTRANSFNS DEFAULTTRANSFNS READFN DONTPRINT*) [Function]

The list of longest translated words in *FILES* is returned.

(SEXPRCOUNT *FILE RDTBL*) [Function]

Returns the number of s-expressions in *FILE* using *RDTBL* to read.

(WORDCOUNT *IFILES COMMONTRANSFNS DEFAULTTRANSFNS READFN DONTPRINT*) [Function]

The total number of translated words in *IFILES* is returned.

(FINDPREFIXES *IFILES OFILE PREFIXES BUTNOT READFN*) [Function]

FINDPREFIXES produces an output file *OFILE* of those strings read by *READFN* from *IFILES* which match at least one prefix in the list of prefix strings *PREFIXES* and do not match any prefixes in the list of prefixstrings *BUTNOT*.

(FINDSUFFIXES *IFILES OFILE SUFFIXES BUTNOT NOCAPS READFN*) [Function]

FINDPREFIXES produces an output file *OFILE* of those strings read by *READFN* from *IFILES* which match at least one suffix in the list of suffix strings *SUFFIXES* and do not match any suffixes in the list of suffix strings *BUTNOT*. If *NOCAPS* is specified then the match succeeds if the string does not have the first letter capitalized.

(FINDSUBSTRINGS *IFILES OFILE SUBSTRINGS READFN*) [Function]

FINDSUBSTRINGS produces an output file *OFILE* of those strings read by *READFN* from *IFILES* which match at least one substring in the list of substrings *SUBSTRINGS*.

Translation Functions

(MIXEDCASEP *W*) [Function]

Returns W if it contains mixtures of uppercase and lowercase characters after the initial character.

(PROPERP *W*) [Function]

Returns T if the first characters of W is uppercase.

(NOTPROPERP *W*) [Function]

Returns T if the first characters of W is not uppercase.

(REVERSESTRING *W STR*) [Function]

Reverses W into STR and returns STR.

Examples

This example will printout all the lines that have either the prefix "re" or "no" but not the prefix "non".

```
(FINDPREFIXES  '{dsk}Myfile T '("re" "no") '("non") (FUNCTION DREADLINE))
```

This example will output to file {Phylum}<Project>Suffixes all the words in the files {dsk}File1 and {dsk}File2 that end in "ion" that do not have the first letter capitalized.

```
(FINDSUFFIXES  '({dsk}File1 {dsk}File2) '{Phylum}<Project>Suffixes "ion" NIL
T (FUNCTION DREADWORD))
```

Miscellaneous Functions

(GETNTHFIELD *STRING N SEPARATOR FIELDTYPE*) [Function]

The Nth field in STRING is returned using *SEPARATOR* as the field separator and the field type is coerced to type *FIELDTYPE*.

N is a simple positive integer [Argument]

SEPARATOR is the same as that for SORTWORDFILE [Argument]

FIELDTYPE is either the atom NUMBER or the atom STRING and indicates how the type that the field should be coerced to. The default *FIELDTYPE* is STRING.

Example

(GETNTHFIELD "So long and thanks for all the fish" 4 ’WHITESPACE ’STRING)

returns the string "thanks".

(GETNTHFIELD "Joe Smith/5551212/12 Pleasant Lane/" 2 "/" ’NUMBER)

returns the integer 5551212.

(DCOPYSTRING *W STR*) [Function]

Copies string *W* into string *STR* and returns *STR*.

(NUMORDER *NUMBER1 NUMBER2*) [Function]

Returns >>??<<

Example

```
(SETQ  MyString (ALLOCSTRING 1))

(DCOPYSTRING "This is a much longer string" MyString)
```