COMPARETEXT

By Mike Sannella. Tested in Medley by Larry Masinter, updated by Ron Kaplan 12/2021

Uses TEDIT, GRAPHER, REGIONMANAGER

INTRODUCTION

COMPARETEXT is a rather non-standard text file comparison program which tries to address two problems: (1) the problem of detecting certain types of changes, such as detecting when a paragraph is moved to a different part of a document; and (2) the problem of showing the user what changes have been made in a document.

The text comparison algorithm is an adaptation of the one described in the article "A Technique for Isolating Differences Between Files" by Paul Heckel, in CACM, V21, #4, April 1978. The main idea is to break each of the two text files into "chunks" (words, lines, paragraphs, ...), hash each chunk into a hash value, and match up chunks with the same hash value in the two files. This method detects switching two chunks, or moving a chunk anywhere else in the document.

COMPARING TEXT FILES

Two text files can be compared with the following function:

(COMPARETEXT *FILE1 FILE2 HASH.TYPE CHUNKREGION FILELABELS TITLE TEXTWIDTH TEXTHEIGHT*) [Function]

FILE1 and *FILE2* are the names of the two files to compare. The order is not important, except that in the resulting graph the *FILE1* information will appear on the left, and the *FILE2* info on the right. The files may also be provide as input streams.

HASH.TYPE determines how "chunks" of text are defined; how fine-grained the comparison will be. This can be PARA to hash by paragraphs (delimited by two consecutive CRs), LINE to hash by lines (delimited by one CR), or WORD to hash words (delimited by any white space). *HASH.TYPE*=NIL defaults to PARA.

CHUNKREGION is the region on the display screen used for the file comparison graph, the chunk window. If *CHUNKREGION*=NIL, the system asks the user to specify a region, prompting with a region that is just wide enough for the graph. If *CHUNKREGION*=T, a region in the lower left corner is used. IF *CHUNKREGION* is a position, the chunkwindow will be located relative to that position, with its horizontal midpoint at the specified XCOORD and its top at the YCOORD.

FILELABELS is an optional pair of labels that will appear over the columns of the difference graph instead of the (often overly long) full names of the files.

TITLE is an optional title to be used for the comparison window.

TEXTWIDTH and *TEXTHEIGHT* are optional parameters that control the size of each of the two text-display windows.

COMPARETEXT creates a graph with two columns. Each column contains the label for one of the files, and lists the chunks from that file. Each chunk is represented by an atom NNN:MMM, where NNN is the file pointer of the beginning of the chunk within the file, and MMM is the length of the chunk. Lines are drawn from one column to the other to show which chunks in one file are the same as those in the other file. Chunks with no lines going to them do not exist in the other file. [Note: a series of chunks in one file which are the same as a series of chunks in the other file are merged into one big chunk. A series of unconnected chunks is also merged.]

Pressing the LEFT mouse button over one of the chunk nodes causes the node and its counterpart in the other column to be inverted, and read-only Tedit windows are open on the files with the appropriate text selected. If a Tedit window to a file is already active, the selection is simply moved. If COMPARETEXT.AUTOTEDIT is true (initially), then regions are selected automatically for the Tedit windows, otherwise the mouse must be used to specify ghost regions.

Pressing the MIDDLE mouse button over a chunk node raises a pop-up menu with the items: PARA, LINE, and WORD. If one of these is selected, COMPARETEXT is called to compare the selected chunk with the last selected chunks (the ones that are boxed), using the hash type selected, and create a new graph window.

White space (space, tab, CR, LF) is used to delimit chunks, but is ignored when computing the hash value of a chunk. Therefore, if two paragraphs are identical except that one has a few extra CRs after it, they will be considered identical by COMPARETEXT.

If the variable COMPARETEXT.ALLCHUNKS is NIL (initially T), then the graph is abbreviated so that nodes for identical chunks in the same position are not shown.