README

The program builds Automat and implements it using a linked list in the following form:

Build the Automat: Get words and add to the Automat

Word Search: Get a word and search for what words that are inside the word exist in the Automat

Main function :

The function **pm_init**: rebooting Automat.

The function **pm_addstring** : adds words to automaton by using auxiliary functions , (pm_goto_get) one that searches if a certain rib has already existed and (pm_goto_set) one that adds a rib if we did not find the rib.

given that the string is of length n. Returns 0 on success, -1 on failure.

The function **pm_goto_set**: Set a transition arrow from this from_state, via a symbol, to a to_state. will be used in the

pm_addstring and pm_makeFSM functions

Returns 0 on success, -1 on failure.

The function **pm_goto_get**: Returns the transition state. If no such state exists, pm_makeFSM, pm_fsm_search, returns NULL will be used in pm_addstring, pm_destroy functions

The function **pm_makeFSM**: Finalizes construction by setting up the failrue transitions, as well as the goto transitions of the zerostate Returns 0 on success, -1 on failure.

The function **pm_fsm_search**: Search for matches in a string of size n in the FSM if there are no matches return empty list

The function **pm_destroy**: A function that calls a recursive function and with its help releases the automaton.

The function **rec_destroy**: Recursive function that frees the mallocs,

Linked List Functions:

slist_init, slist_destroy, slist_pop_first, slist_append, slist_prepend, slist_append_list.

Program file:

slist.c , slist.h, pattern_matching.c, pattern_matching.