

Objective: Development of a simple handmade Lexical analyzer (NFA and DFA Simulator)

This is an extension to your previous Lab sessions. As discussed in the class, your simulator should be able to accept the state transition table of any given NFA or DFA, then it should verify whether the input string(s) given through the file *strings.txt* are acceptable (in which case print "yes") or not (print "no").

**Example 1: ./slex 4 1 2 0**

The fourth argument distinguishes DFA and NFA. 0 for NFA 1 for DFA.

These command line arguments represent the details of an NFA with 4 states with a single final state and the alphabet has 2 symbols.

The other specifications of NFA that go into the file *table.txt* are the following,

0 1 2 3 // num of states 4, each state is separated by space

3 // final state is state-3

a b // symbols in alphabet

The transition table is a matrix of 4x3, in this case (2 symbols from alphabet and \$ which represents epsilon). ^ represents null set.

	a	b	\$
0	0,1	0	^
1	^	2	^
2	^	3	^
3	^	^	^

This NFA accepts the strings of a,b ending with abb.

**Example 2: ./mylex 5 2 2 0**

Represents an NFA with 5 states of which 3 is a final state and the alphabet has 2 symbols.

The other specifications of NFA that go into the file *table.txt* are the following,

0 1 2 3 4 // num of states 5, each state is separated by space

2 4 // final states are state-2 and state-4

a b // symbols in alphabet

The transition table in a matrix of 5x3, in this case (2 symbols from alphabet and \$ which represents epsilon). ^ represents null set.

	a	b	\$
0	^	^	1,3
1	2	^	^
2	2	^	^
3	^	4	^
4	^	4	^

This NFA accepts the strings a, b, aa, bb, aaa, bbb, .

Submission instructions: The name of the executable is **slex** and other instructions are same as those of previous exercises.