Task: 1
Turing machine simulation

deadline: Apr 8, 2018, 23:55

Consider the following problem: does the given non-deterministic Turing machine M accept a given word w after at most $|w|$ steps without ever leaving the cells initially occupied by w? The machine is presented as $M = (\Sigma, Q, q_I, q_F, \delta)$, where $\Sigma = \{a_1, \ldots, a_k\}$ is the alphabet, $Q = \{q_1, \ldots, q_n\}$ is the set of states, and $\delta \subseteq \Sigma \times Q \times \Sigma \times Q \times \{-1, 0, 1\}$ is the set of transitions. We assume the machine starts with the head over the leftmost symbol in q_I, and accepts when it reaches q_F.

Write a program that reads the machine and the word from standard input (in the format described below), translates the problem above to SAT, runs Z3 over it, and writes to standard output either `NO`, or `YES` followed by an accepting run (in the format described below).

Input

The first line of the input contains three numbers: k (the number of symbols), n (the number of states), and m (the number of transitions). The second line of the input contains k distinct words $a_1, \ldots, a_k$ separated by single spaces, representing the symbols of the alphabet. The third line of the input contains n distinct words $q_1, \ldots, q_n$ separated by single spaces, representing the states of the machine.

The fourth line contains a single word q_I representing the initial state, and the fifth line contains a single word q_F representing the final state. Both words q_I and q_F appear in the list of states (the third line).

The next m lines represent the transitions. A transition is represented as $a_c\ q_c\ a_n\ q_n\ d$. Such a transition can be executed if the current state of the machine is q_c, and the symbol below the head is a_c. The machine changes its state to q_n, the symbol below the head to a_n, and moves the head according to d. The value of d could be -1 (move to the left), 0 (stay in place) or 1 (move to the right). For each transition, the words q_c and q_n appear in the list of states (the third line), and the words a_c and a_n appear in the list of symbols (the second line).

The next line contains a single number l (the length of the word). The next line contains l words $w_1, \ldots, w_l$ separated by single spaces, representing the input word. Each w_i appears in the list of symbols (the second line).

Output

In case if no accepting run has been found, the first and only line of the output should contain `NO`. Otherwise, the first line should contain `YES`, the second line should be empty, and the following lines should list the configurations of the machine during the run. Each configuration is a sequence of cell contents separated by a single space, each cell content is written as symbol | state, and state is a single space if there is no state in this cell.

Example

For the input data:

```
3 2 2
a b c
i f
i
f
a i a i 1
b i a f 0
4
a a b a
```

the correct result is:

```
YES

a|i a|  b|  a|
a|  a|i b|  a|
a|  a|  b|i a|
a|  a|  a|f a|
```

Allowed languages

You are allowed to choose any programming language accepted by the graders. The following languages are accepted: C, C++, Java, Python, OCaml, Haskell. Other languages need approval of the graders (Lorenzo Clemente and Filip Murlak). You can also choose to either use Z3 directly via the API, or to communicate with Z3 via the DIMACS format.