

# ANOTHER TURING MACHINE SIMULATOR, IMPLEMENTED AS A JAVA(tm) APPLICATION

Version 1.2 (November 1, 1997)

(tm) Java is a registered trademark of Sun Microsystems

This is a simulator for *deterministic* Turing machines, following the style in J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979. A simulator for *non-deterministic* Turing machines is being planned and will appear after the next project, which is a simulator for a non-deterministic push-down automaton.

Please note that there is also a [finite-state machine simulator](#) available

## Contents:

- [Obtaining the Java interpreter](#)
- [Obtaining the Turing machine simulator](#)
- [Setting the environment](#)
- [Obtaining the example files](#)
- [Using the graphical version of the Simulator](#)
- [The course-grader's quick-to-use version of the Simulator](#)
- [The short-form of the text-only version of the Simulator](#)
- [The full text-only version of the Turing machine simulator](#)
- [Format of the Turing Machine file](#)
  - [The blank symbol](#)
  - [Example 1](#)
  - [Example 2](#)
  - [Example 3](#)
  - [Example 4](#)

---

## Obtaining the Java interpreter

Down-load the Java Development Kit using a browser (lynx, Internet explorer, Netscape), go to the site <http://www.javasoft.com> and select "Java Development Kit." You need to download the JDK1.1.4 (or later) for your machine, probably Microsoft's Windows 95/NT, although the simulator also works on many UNIX platforms. If you are running Windows 3.1, you must go to the IBM site and look there: <http://www.alphaworks.ibm.com> (this site does not work usefully with lynx). For Windows 3.1 you are looking for IBM's ADK for Java 1.1.

Be prepared for several hours download time (try overnight).

The file is a self-extracting executable. It should install into the directory C:\jdk1.1.4 (check the installation notes if you are using IBM's ADK).

If you plan to develop programs in Java, download the "api documentation" and **unzip it using a 32-bit (long file name) unzipper**. Unzip to C:\ with directory options on; in this way it unzips to C:\jdk1.1.4\docs (or later). Also download the Java tutorial when you can (let me know if you cannot find the tutorial.zip file). The tutorial has to be unzipped into the directory it will use, e.g., C:\jdk1.1.4\tutorial.

---

## Obtaining the Turing machine simulator

*No file* copying is necessary if you are working locally and using our server bingsuns.

For your own PC, download the file [tmsim\\_bgm.zip](#) or [tmsim\\_bgm.tar](#) and unzip from your root directory. We recommend that you keep the files in the directories shown and add C:\classes your CLASSPATH variable, *despite* the current recommendation from Javasoft not to use a CLASSPATH. The files provided are:

(the path shown is for WINDOWS, you get the equivalent on a UNIX machine)

C:\classes\DetailText.class

C:\classes\Grader.class

C:\classes\QuickText.class

C:\classes\TM.class

C:\classes\ATuring\CButton.class

C:\classes\ATuring\Controller\$1.class

C:\classes\ATuring\Controller\$2.class

C:\classes\ATuring\Controller\$3.class

C:\classes\ATuring\Controller\$4.class

C:\classes\ATuring\Controller\$5.class

C:\classes\ATuring\Controller.class

C:\classes\ATuring\CreatTM.class

C:\classes\ATuring\Key.class

C:\classes\ATuring\MachineDescription.class

C:\classes\ATuring\MachinePanel\$1.class

C:\classes\ATuring\MachinePanel\$InputActionListener.class

C:\classes\ATuring\MachinePanel.class

C:\classes\ATuring>MainFrame\$1.class

C:\classes\ATuring>MainFrame\$2.class

C:\classes\ATuring>MainFrame\$3.class

C:\classes\ATuring\MainFrame\$4.class  
C:\classes\ATuring\MainFrame.class  
C:\classes\ATuring\RunMachine.class  
C:\classes\ATuring\Set.class  
C:\classes\ATuring\State.class  
C:\classes\ATuring\Tape.class  
C:\classes\ATuring\TapeCanvas.class  
C:\classes\ATuring\TextDriver\$InputData.class  
C:\classes\ATuring\TextDriver.class  
C:\classes\ATuring\Transition.class  
C:\classes\ATuring\TuringMachine.class  
C:\classes\ATuring\TuringMachineException.class  
C:\classes\ATuring\Viewer.class  
C:\classes\ATuring\XQTThread.class  
C:\tm\_source\tm1.txt  
C:\tm\_source\tm2.txt  
C:\tm\_source\tm3.txt  
C:\tm\_source\tm4.txt  
C:\tm\_source\tm5.txt  
C:\tm\_source\tm6.txt  
C:\tm\_source\tm7.txt  
C:\tm\_source\tm8.txt  
C:\tm\_source\tmdoc.html  
C:\tm\_source\tmfig1.gif  
C:\tm\_source\tmfig10.gif  
C:\tm\_source\tmfig11.gif  
C:\tm\_source\tmfig12.gif  
C:\tm\_source\tmfig2.gif  
C:\tm\_source\tmfig3.gif  
C:\tm\_source\tmfig4.gif  
C:\tm\_source\tmfig5.gif  
C:\tm\_source\tmfig6.gif  
C:\tm\_source\tmfig7.gif  
C:\tm\_source\tmfig8.gif  
C:\tm\_source\tmfig9.gif  
C:\tm\_source\input1.txt  
C:\tm\_source\input3.txt  
C:\tm\_source\input4.txt  
C:\tm\_source\input5.txt  
C:\tm\_source\input6.txt

This list is from the Windows95 environment, the UNIX versions will be similar.

Note that Java is case sensitive, *even on directory names*.

---

## Setting the environment

Open or create AUTOEXEC.BAT.

Add the following two lines after any other modification to the path

```
set path = %path%;c:\jdk1.1.4\bin
```

```
set CLASSPATH = c:\classes
```

Despite suggestions in the Java installation files, we find this CLASSPATH very convenient. On the server bingsuns you have to modify the file ".cshrc" to add the lines (check which is the latest jdk)

```
setenv PATH ${PATH}:/opt/local/java/jdk1.1.4/bin
```

```
setenv CLASSPATH u0/users/A/head/classes:<your login path>/classes
```

---

## Obtaining the example files

The files obtained will be (you can also save them from the following links) [tm1.txt](#), [tm2.txt](#), [tm3.txt](#), [tm4.txt](#), [tm5.txt](#), [tm6.txt](#), [tm7.txt](#), [tm8.txt](#). The languages that these machines accept are annotated in the heading of the machine text and as the title of the Java frame that runs the simulation. When you download these files make sure there are no empty lines at the top of the file. They are included in the zip and tar files and will be placed in a subdirectory C:\tm\_source. The test files [input1.txt](#), [input3.txt](#), [input4.txt](#), [input5.txt](#) and [input6.txt](#) are used in conjunction with the text-only versions of the program, as described below.

---

## Using the graphical version of the Simulator

When you type in the command window (MS-DOS window on Windows 95/NT and an X-Term window on Solaris)

- java TM

you will see the frame shown in [Figure 1](#).

If you click on "Machine" in the Menu Bar, you pull down the options "New Machine" and

"Restart," as shown in [Figure 2](#).

When you click on "New Machine," you get a File Dialog Box, which allows you to open a file that contains a description of the Turing machine that you are about to test, see [Figure 3](#).

You select and open the appropriate file. The format for the file is described later. The simulator will load and display the tape of the machine and the initial state, see [Figure 4](#). If there are detectable errors in the format of your machine these will be reported in your command window.

Next you should insert an input string. If you do not, the input will be assumed to be the empty string. To insert an input string, click in the empty text field next to the button "Accept Input Tape." Type in the input string, [Figure 5](#), and click the button OR hit enter, [Figure 6](#).

The control bar allows you to proceed *one step at a time* (by pressing "Step") or *run continuously at different speeds* (by pressing one of the smaller buttons, which will show up red). See [Figure 7](#).

## NOTES

- The blank symbol does not show on the tapes of the graphical simulation but it is expressed by the symbol "B" when it appears in a transition in the file describing the Turing Machine, e.g. the transition  $q_3 \text{ B } q_4 \text{ B R}$  in Example 1.
- The simulator has special feature that allows the machine to run *in reverse*, thereby undoing all the transitions that have occurred. Reverse execution *is not possible* after the machine halts (either in an accepting or a non-accepting state).

The number of transition steps executed so far is shown at the top right of the frame.

This message is replaced with an "ACCEPTED" or "Not accepted" message when the machine halts. [Figure 7](#). shows the machine running, [Figure 8](#) shows the machine when it accepts the input and [Figure 9](#) shows the machine when it fails to accept an input. Notice that in [Figure 8](#) and [Figure 9](#), the command window also shows a final status message.

The Turing machine also allows you to introduce a "New Input," which is selected under "Options" on the menu bar, and to "Restart" the machine *with the same input*, which is selected under "Machine" on the menu bar. See [Figure 10](#) and [Figure 11](#)

The machine can handle 1-way infinite tapes, 2-way infinite tapes, multiple tapes, multiple tracks and compound states. [Figure 12](#) shows the Turing machine simulator with several tapes and tracks.

When a course grader is presented with a large number of Turing Machine text files to check, one time saving approach is to run all the machines against a set of test inputs. The grader should prepare a file with a number of test cases, for example the file [input1.txt](#) shown below. The first line (300) is an upper limit on the number of steps that the simulation can run for [Turing machines may run for ever]. The second line (18) is the total number of test cases in the file. Lines 3 through 20 are the test cases. Line 6 contains a "B" representing the blank symbol, which means the input string is empty.

```
300
18
01
00001111
000000000000111111111111
B
11110101111110
11110101111110111110
101010
1010100
10101010
000
0000
1010101010
101010100
11111111
0
00
00000
0110111
```

To run the grader's version, type the following:

```
java Grader
```

or

```
java Grader input1.txt
```

or

```
java Grader input1.txt tm1.txt
```

If the command-line parameters are not provided, there will be prompts to ask for them.

In the example below, the grader is working in the directory "tm\_source" and types in the name of the common set of test cases in the file [input1.txt](#). Then there is a prompt for the next Turing Machine to test.

In this example, the machine file [tm1.txt](#) was used. The following lines give the test results:

"YES" means the current input was accepted and "TuringMachineException, ..." indicates that the simulation was abandoned because the necessary transition was not defined (this possibility is quite normal and means that the Turing machine rejects the input). After checking whether the tests match the correct responses, the program asks for another Turing Machine file to test

(against the same test inputs) so that the grader can run through all the solutions to a specific problem.

The Grader program does not provide useful information for those examples where the Turing Machine is being used to compute a value (e.g. [tm6.txt](#) computes the product of two numbers). It is more convenient to use [QuickText](#) in this case.

```
C:\tm_source>java Grader
Enter Input File Name: input1.txt

TITLE: NEW MACHINE
Enter Input File Name: input1.txt

TITLE: NEW MACHINE
Enter TuringMachine FileName: tm1.txt
INPUT: 000111    YES
INPUT: 00000000001111111111    YES
INPUT: 01    YES
INPUT: B    TuringMachineException:    No transition for q0
INPUT: 11110101111110    TuringMachineException:    No transition for q0
INPUT: 11110101111110111110    TuringMachineException:    No transition for q0
INPUT: 101010    TuringMachineException:    No transition for q0
INPUT: 1010100    TuringMachineException:    No transition for q0
INPUT: 10101010    TuringMachineException:    No transition for q0
INPUT: 000    TuringMachineException:    No transition for q1
INPUT: 0000    TuringMachineException:    No transition for q1
INPUT: 1010101010    TuringMachineException:    No transition for q0
INPUT: 101010100    TuringMachineException:    No transition for q0
INPUT: 11111111    TuringMachineException:    No transition for q0
INPUT: 0    TuringMachineException:    No transition for q1
INPUT: 00    TuringMachineException:    No transition for q1
INPUT: 00000    TuringMachineException:    No transition for q1
INPUT: 0001111    TuringMachineException:    No transition for q3

Another Machine? Y/N
```

---

## **The short-form of the text-only version of the Simulator**

A second text-only versions is designed to let a student work at an ASCII terminal, typically through a dial-up connection, which normally prevents the use of the GUI version. To run the simpler version, simply type one of the following:

```
java QuickText
```

or

```
java QuickText input1.txt
```

or

```
java QuickText input1.txt tm1.txt
```

or

```
java QuickText input1.txt tm1.txt log1.txt
```

If the parameter "log1.txt" is not provided, the default is "report.txt." If the other parameters are not provided, there will be prompts to request them. The file of test cases [input1.txt](#) is the same as before. The following example shows the use of the same Turing Machine [tm1.txt](#), which was used above.

```
C:\tm_source>java QuickText
Enter Input File Name: input1.txt

TITLE: NEW MACHINE
Enter TuringMachine FileName: tm1.txt

Ready? return
```

The program reports on whether the Turing Machine accepted a particular input from the file [input1.txt](#), in this case. The reports come in groups of 5. Examples include the following (the second example is a test with a empty input, note that "T/T" means "tapes/tracks" and "0/0" means "tape 0/track 0"):

```
TITLE: NEW MACHINE

TITLE MACHINE: Version of Hopcroft & Ullman's TM for {0^n 1^n : n > 0} with 2-way
INPUT: 000111
STATUS: Version of Hopcroft & Ullman's TM for {0^n 1^n : n > 0} with 2-way infin
STATUS: FINAL T/T: (0/0) BBBBBBBBBBBBBBXXXYYB<read-head>BBBBBBBBBBBBBBBBBBBB

TITLE MACHINE: Version of Hopcroft & Ullman's TM for {0^n 1^n : n > 0} with 2-way
INPUT: B
TuringMachineException: No transition for q0
STATUS: Execution terminated Error in Version of Hopcroft & Ullman's TM for {0
STATUS: FINAL T/T: (0/0) BBBBBBBBBBBBBB<read-head>BBBBBBBBBBBBBBBBBBBB

TITLE MACHINE: Version of Hopcroft & Ullman's TM for {0^n 1^n : n > 0} with 2-way
INPUT: 0001111
TuringMachineException: No transition for q3
STATUS: Execution terminated Error in Version of Hopcroft & Ullman's TM for {0
STATUS: FINAL T/T: (0/0) BBBBBBBBBBBBBB<read-head>1BBBBBBBBBBBBBBBBBBBB
```

The same information shown on the screen is saved in a slightly different format in the log file (default name: report.txt).

**Note** that the symbol "B" is used to denote a blank symbol on the tape.

---

## The full text-only version of the Turing machine simulator

For the text-only version, the Turing Machine files are prepared in the same way as for the graphical version of the simulator. In this case, as before you need to prepare a file of inputs such



as [input1.txt](#). The file was described in the section on the [Grader's version](#) of the program. The results of the tests will be written to an output log, which has a default name is "report.txt" but that can be modified by the user. This text-only version can be run using one of the following choices:

```
java DetailText
```

or

```
java DetailText input1.txt
```

or

```
java DetailText input1.txt tm1.txt
```

or

```
java DetailText input1.txt tm1.txt log1.txt
```

If the parameter "log1.txt" is not provided, the default is "report.txt." If the other parameters are not provided, there will be prompts to request them. On the screen the results of each test are shown in exactly the same way as for the QuickText program described above(accept/reject on account of a missing transition) with the name of the machine, the test input, the state where execution ended. The big difference is in the log file. Note that in this version, after each run of the simulator, you may also change the Turing Machine, keeping the same test file (this feature is provided specifically for the grader who might be diagnosing the failure of students' work).

The format of the log file is shown in the following examples taken from [tm1.txt](#) and [input1.txt](#). Each step shows the tape contents. The read/write head is considered to be looking at the tape symbol written immediately after the <read-head> marker.

...

```
TITLE  MACHINE: Version of Hopcroft & Ullman's TM for {0^n 1^n : n > 0} with 2-way
      INPUT: 000111
SNAPSHOT:  START State:q0
SNAPSHOT:  Machine Tapes:
SNAPSHOT: (Tape,Track):(00) BBBBBBBBBB<read-head>000111BBBB
SNAPSHOT:

SNAPSHOT: Step number: 1,  Current State: q1
SNAPSHOT: (Tape,Track):(00) BBBBBBBBBX<read-head>00111BBBBB

SNAPSHOT: Step number: 2,  Current State: q1
SNAPSHOT: (Tape,Track):(00) BBBBBBBBBX0<read-head>0111BBBBBB
```

...

```

SNAPSHOT: Step number: 24, Current State: q3
SNAPSHOT: (Tape,Track):(00) BBBBXXXYYY<read-head>BBBBBBBBBBB

SNAPSHOT: Step number: 25, Current State: q4
SNAPSHOT: (Tape,Track):(00) BBBXXXYYB<read-head>BBBBBBBBBBB

SNAPSHOT: Stopped ACCEPTED after Step: 25
SNAPSHOT: FINAL TAPE(S):
SNAPSHOT: (Tape,Track):(00) BBBBBBBBBBBBBBXXXYYB<read-head>BBBBBBBBBBBBBBBBBBBB

```

From [tm5.txt](#) and [input5.txt](#), we show two of the steps. Note that just a single step covers several lines; each track of each tape is listed separately. Note that one-way infinite tapes may have fewer (or no) symbols to the left of the read/write head than the two-way infinite tapes, which always have the same number of symbols to the left of the read/write head. In this example, the individual tapes and tracks are identified by the pairs 00, 01, 02, 10, 20, 21, 30, 31.

...

```

SNAPSHOT: Step number: 2, Current State: [q2,axy]
SNAPSHOT: (Tape,Track):(00) BBBBBBBB12<read-head>aaaaaaaaaaa
SNAPSHOT: (Tape,Track):(01) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(02) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(10) BB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(20) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(21) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(30) <read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(31) <read-head>BBBBBBBBBBB

```

...

```

SNAPSHOT: Step number: 15, Current State: [q2,zzz]
SNAPSHOT: (Tape,Track):(00) 4444444444<read-head>aaabBBBBBB
SNAPSHOT: (Tape,Track):(01) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(02) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(10) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(20) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(21) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(30) BBBBBBBBBB<read-head>BBBBBBBBBBB
SNAPSHOT: (Tape,Track):(31) BBBBBBBBBB<read-head>BBBBBBBBBBB

```

**Note** that the symbol "B" is used to denote a blank symbol on the tape.

---

### Format of the Turing Machine file

The format for the Turing machine files is the following. The first line indicates that this is A Turing Machine file. Later versions of the tool will also work with files describing finite state machines and push-down automata. There should be *no blank lines*:

Line 1: ATM.

Line 2: name of the machine. For assignments use name(s) and assignment number

Line 3: all characters in the input alphabet, separated by spaces

Line 4: all characters in the tape alphabet, separated by spaces

Line 5: the number of tapes (N). They are numbered 0,..., N-1

Line 6: the number of tracks on tape 0 ( $T_0$ )

Line 7: the number of tracks on tape 1 (if  $N > 1$ ) ( $T_1$ )

...

Line N+5: the number of tracks on tape N-1 ( $T_{N-1}$ )

Line N+6: 1 or 2, indicating tape 0 is 1-way or 2-way infinite tape

Line N+7: 1 or 2, indicating tape 1 is 1-way or 2-way infinite

...

Line 2N+5: 1 or 2, indicating tape N-1 is 1-way or 2-way infinite

Line 2N+6: the initial state of the Turing machine

Line 2N+7: all the final states separated by spaces

Line 2N+8: first of a list of transition (see below)

Line 2N+9: another transition

...

Line 2N+?: last of the list of transitions

Last line: end. You must type the word "end" but it can be upper or lower case

The format of a transition is:

<state> <content of tracks> <next-state> <next content of tracks> <head-move directions>

where

- the <state> and <next-state> are the names of states (simple strings). When the state has multiple fields such as [q0,a,b], there can be *no spaces*. This compound state is handled in the simulator as a *single string*, without spaces.
- the <content of tracks> and <next content of tracks> have the general form:  
 $X_1+X_2+X_3+\dots+X_L$ , where L is the total number of tracks, i.e. L is the sum of  $T_0, T_1, T_2, \dots, T_{N-1}$  and the  $X_j$  are the symbols on the individual tracks above the read/write head. Actually there is a read/write head for each tape but only one is shown in the graphical simulator.
- the <head-move directions> has the general form  $d_1+d_2+\dots+d_N$ , where each  $d_j$  is R, L or S, indicating a right or left movement or staying stationary of the tape head.

## The Blank Symbol

As shown in the examples below, the blank tape symbol is denoted by "B" and this symbol *must not* be used as part of the input alphabet. The blank symbol is not shown on the tapes of the graphical simulator.

## Example 1:

ATM

```
Hopcroft & Ullman's TM for  $\{0^n 1^n : n > 0\}$  (file name tml.txt)
0 1 // input alphabet, note such comments are permitted at the end of the line
0 1 X Y B // tape alphabet
1 // number of tapes
1 // number of tracks on tape 0
2 // tape 0 is 2-way infinite
q0 // the initial state
q4 // final state
q0 0 q1 X R // the transitions do not use "+" since there is only one track
q0 Y q3 Y R
q1 0 q1 0 R
q1 1 q2 Y L
q1 Y q1 Y R
q2 0 q2 0 L
q2 X q0 X R
q2 Y q2 Y L
q3 Y q3 Y R
q3 B q4 B R
end //required
```

## Example 2:

ATM

```
Hopcroft & Ullman's TM for  $01^* + 10^*$  (file name cs373b.txt)
0 1 // input alphabet
0 1 B // tape alphabet
1 // number of tapes
1 // number of tracks on tape 0
2 // tape 0 is 2-way infinite
[q0,B] // the initial state NOTE THERE CAN BE NO SPACES IN THIS STRING
1 // the number of final states
[q1,B] // the final state
[q0,B] 0 [q1,0] 0 R // the transitions do not use "+" since there is only one track
[q0,B] 1 [q1,1] 1 R //NOTE THERE CAN BE NO SPACES IN ANY OF THE STATES
[q1,0] 1 [q1,0] 1 R //NOTE THERE CAN BE NO SPACES IN ANY OF THE STATES
[q1,1] 0 [q1,1] 0 R //NOTE THERE CAN BE NO SPACES IN ANY OF THE STATES
[q1,0] B [q1,B] 0 L //NOTE THERE CAN BE NO SPACES IN ANY OF THE STATES
[q1,1] B [q1,B] 0 L //NOTE THERE CAN BE NO SPACES IN ANY OF THE STATES
end
```

## Example 3:

ATM

```
Hopcroft & Ullman's TM for  $\{wcw \mid w \text{ in } (a + b)^+\}$  (file name cs373c.txt)
a b // input alphabet
a b T B // tape alphabet
1 // number of tapes
2 // number of tracks on tape 0
```

```

2 // tape 0 is 2-way infinite
[q1,B] // initial state
[q9,B] // the final state, NOTE THERE CAN BE NO SPACES IN ANY OF THE STATES
[q1,B] a+B [q2,a] a+T r // "a+B" means "a" on the top track, "B" on the lower track
[q1,B] b+B [q2,b] b+T r // the single "r" means the tape head moves right
[q2,a] a+B [q2,a] a+B r
[q2,a] b+B [q2,a] b+B r
[q2,b] a+B [q2,b] a+B r
[q2,b] b+B [q2,b] b+B r
[q2,a] c+B [q3,a] c+B r
[q2,b] c+B [q3,b] c+B r
[q3,a] a+T [q3,a] a+T r
[q3,a] b+T [q3,a] b+T r
[q3,b] a+T [q3,b] a+T r
[q3,b] b+T [q3,b] b+T r
[q3,a] a+B [q4,B] a+T l
[q3,b] b+B [q4,B] b+T l
[q4,B] a+T [q4,B] a+T l
[q4,B] b+T [q4,B] b+T l
[q4,B] c+B [q5,B] c+B l
[q5,B] a+B [q6,B] a+B l
[q5,B] b+B [q6,B] b+B l
[q6,B] a+B [q6,B] a+B l
[q6,B] b+B [q6,B] b+B l
[q6,B] a+T [q1,B] a+T r
[q6,B] b+T [q1,B] b+T r
[q5,B] a+T [q7,B] a+T r
[q5,B] b+T [q7,B] b+T r
[q7,B] c+B [q8,B] c+B r
[q8,B] a+T [q8,B] a+T r
[q8,B] b+T [q8,B] b+T r
[q8,B] B+B [q9,B] B+T l
end

```

## Example 4:

ATM

An example of multiple tapes (cs373d.txt). It recognizes  $a^3a^*b$  (only the top track

```

a b // input alphabet
a b B // tape alphabet
4 // number of tapes
3 // number of tracks on tape 0
1 // number of tracks on tape 1
2 // number of tracks on tape 2
2 // number of tracks on tape 3
2 // tape 0 is 2-way infinite
1 // tape 1 is 1-way infinite
2 // tape 2 is 2-way infinite
1 // tape 3 is 1-way infinite
[q0,B] // the initial state
[q1,B] [q2.1] [q3,X] // final states
[q0,B] a+B+B+B+B+B+B+B [q1,ab] 1+B+B+B+B+B+B+B R+R+L+S // "S" means stationary
[q1,ab] a+B+B+B+B+B+B+B [q2,axy] 2+B+B+B+B+B+B+B R+R+L+S

```

[q2,axy]	a+B+B+B+B+B+B+B	[q2,zzz]	3+B+B+B+B+B+B+B	R+R+L+R
[q2,zzz]	a+B+B+B+B+B+B+B	[q2,zzz]	4+B+B+B+B+B+B+B	R+R+L+R
[q2,zzz]	b+B+B+B+B+B+B+B	[q3,X]	5+B+B+B+B+B+B+B	L+R+L+S

eNd