CS-311 Project#1 FA

Team members

Description

An universal FA is implemented in Java with the following features:

- DFA w/ or w/o trap state
- Alphabet group (ex. all digits[0-9] wraped in one group which acts like a single alphabet in state transition) — this is for M3 and M4
- Contruct a specific FA from definition file with key information provided (Alphabet set, State set, Initial state, Final states, Transition table)
- Test cases are also included in the definition file

Understand the directoies

```
— < project root >
 |— classes          —> Complied binary files go here
 |— data             —> Definition files
 |— source           —> Source code
 |— doc              —> document files
 |— README.md
 |— compile.sh
```

How to compile

Linux/Mac OSX

```
cd < project root >
sh compile.sh
```

Windows

Unfortunately we didn't write a batch file to do the compile work, but you can copy/paste the commands in compile.sh and execute them to get the code compiled.

Notes: Code only got tested in Java 1.6! Since we don't leverage any "fancy" functionalities beyond standard Java libary, you should be ok to compile the code in any version greater than 1.6

How to run

```
cd classes
java FiniteAutomataApplication ../data/<definition file name>
```

M1

Description

An FA which recognizes the set of strings over {0,1} that ends with 0.

Input file

data/FA1.txt

Output

```
Liang@MBP classes $ java FiniteAutomataApplication ../data/FA1.txt
FA Definition:
AlphabetList = {
        0 -> 0
        1 -> 1
}
StateList = {0,1}
InitState = 0
FinalStateList = {1}
TransitionTable = {
        (0, 0, 1)
        (0, 1, 0)
        (1, 0, 1)
        (1, 1, 0)
}

Test Cases:
 "" --> reject
 "100" --> accept
 "011" --> reject
 "10abc" --> reject
 "0" --> accept
 "1" --> reject
 "0101011" --> reject
 "11010" --> accept
 "0001" --> reject
 "1110" --> accept
```

M2

Description

An FA which recognizes the set of strings over {0,1} that do not have two consecutive 1’s.

Input file

data/FA2.txt

Output

```
Liang@MBP classes $ java FiniteAutomataApplication ../data/FA2.txt
FA Definition:
AlphabetList = {
        0 -> 0
        1 -> 1
}
StateList = {0,1,2}
InitState = 0
FinalStateList = {1,2}
TransitionTable = {
        (2, 0, 1)
        (0, 0, 1)
        (0, 1, 2)
        (1, 0, 1)
        (1, 1, 2)
}

Test Cases:
 "" --> reject
 "1" --> accept
 "000" --> accept
 "101" --> accept
 "111" --> reject
 "01001" --> accept
 "1011011" --> reject
 "1011000" --> reject
 "01010" --> accept
 "1010101110" --> reject
```

M3

Description

An FA which recognizes all identifiers that begin with a letter (both upper and lower) and followed by any number of letters and digits.

Input file

data/FA3.txt

Output

```
Liang@MBP classes $ java FiniteAutomataApplication ../data/FA3.txt
FA Definition:
AlphabetList = {
        0 ->
a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z
        1 -> 0,1,2,3,4,5,6,7,8,9
}
StateList = {0,1}
InitState = 0
FinalStateList = {1}
TransitionTable = {
        (1, 0, 1)
        (0, 0, 1)
        (1, 1, 1)
}

Test Cases:
 "" --> reject
 "HelloWorld" --> accept
 "abc" --> accept
 "1st_Ex" --> reject
 "Java" --> accept
 "finite_automaton" --> reject
 "program" --> accept
 "X3Y7" --> accept
 "X=90" --> reject
 "X*Y" --> reject
```

M4

Description

An FA which recognizes the set of all decimal unsigned integer numbers without leading zeros except the number 0 (i.e. number 0 should be accepted while number 01 should be rejected.)

Input file

data/FA4.txt

Output

```
Liang@MBP classes $ java FiniteAutomataApplication ../data/FA4.txt
FA Definition:
AlphabetList = {
        0 -> 0
        1 -> 1,2,3,4,5,6,7,8,9
}
StateList = {0,1,2,3}
InitState = 0
FinalStateList = {1,2}
TransitionTable = {
        (2, 0, 2)
        (0, 1, 2)
        (0, 0, 1)
        (2, 1, 2)
        (1, 1, 3)
        (1, 0, 3)
}

Test Cases:
 "7" --> accept
 "-7" --> reject
 "007" --> reject
 "3.14" --> reject
 "103" --> accept
 "24930000" --> accept
 "0" --> accept
 "01" --> reject
 "100" --> accept
 "0101" --> reject
```