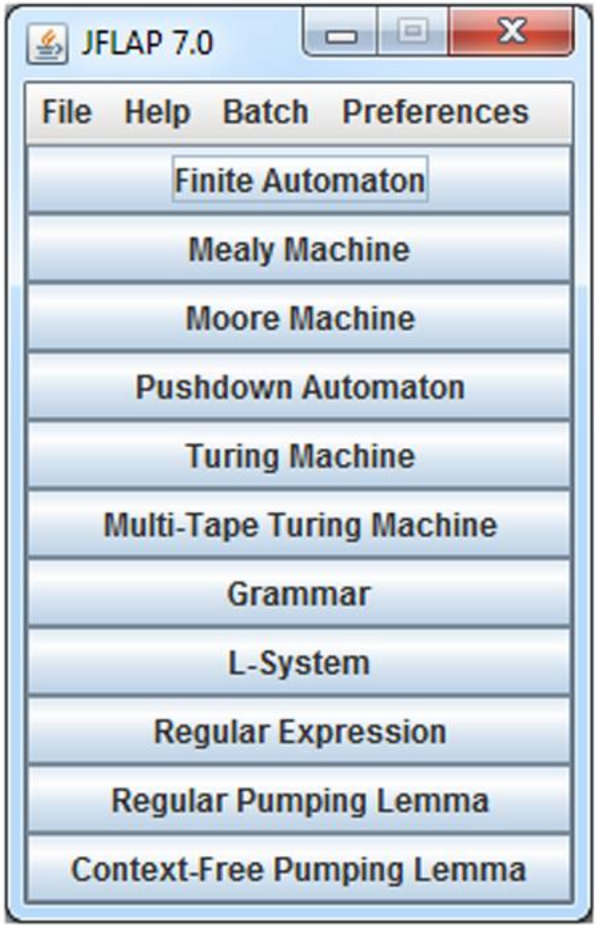
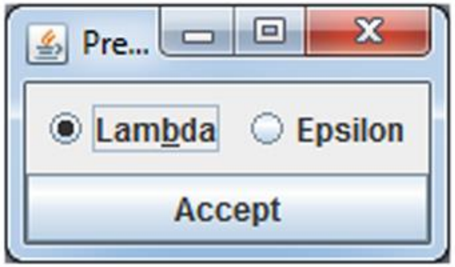
Finite Automata Theory and Formal Languages

(Week 2, Lecture 2)

JFlap





● Lambda ○ Epsilon

Demo

Implementing FA (Scheme 1)

Non-Final State with two alphabets:

```
IF position=length_of_string THEN
        PRINT "String not accepted“ and quit_program

READ current_character from the string and INC counter

IF current_character=alphabet1      THEN CALL proc/funct
ELSE IF current_character=alphabet2 THEN CALL proc/funct
ELSE PRINT "Invalid Alphabets in String“ and quit_program
```

Implementing FA (Scheme 1)

Final State with two alphabets:

```
IF position=length_of_string THEN
        PRINT "String accepted“ and quit_program

READ current_character from the string and INC counter

IF current_character=alphabet1      THEN CALL proc/funct
ELSE IF current_character=alphabet2 THEN CALL proc/funct
ELSE PRINT "Invalid Alphabets in String“ and quit_program
```

Implementing FA (Scheme 1)

Runner:

```
READ string
CALL initial_state_proc/func
```

- ISSUE?

Implementing FA (Scheme 2)

Non-Final State with two alphabets:

```
IF position=length_of_string THEN
            PRINT "String not accepted“ and quit_program

READ current_character from the string and INC counter

IF current_character=alphabet1          THEN SET state_variable
ELSE IF current_character=alphabet2 THEN SET state_variable
ELSE PRINT "Invalid Alphabets in String“ and quit_program
```

Implementing FA (Scheme 2)

Final State with two alphabets:

```
IF position=length_of_string THEN
        PRINT "String accepted“ and quit_program

READ current_character from the string and INC counter

IF current_character=alphabet1      THEN SET state_variable
ELSE IF current_character=alphabet2 THEN SET state_variable
ELSE PRINT "Invalid Alphabets in String“ and quit_program
```

Implementing FA (Scheme 2)

Runner:

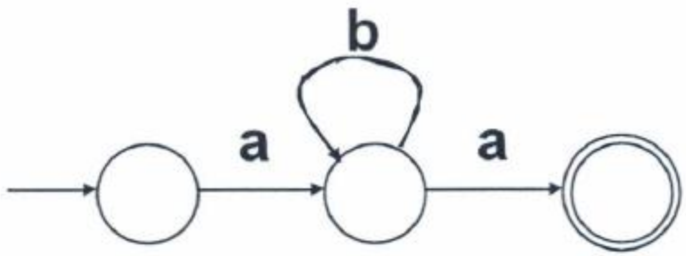
```
READ string
SET state_variable to initial

LOOP: CALL proc/func on the basis of state_variable
```

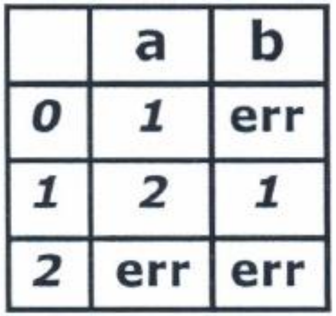
- ISSUE?

Implementing FA (Scheme 3)

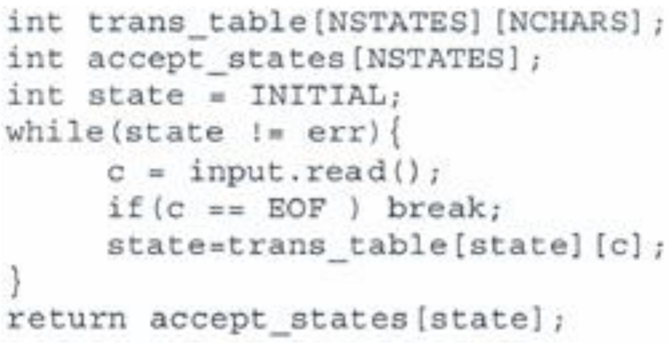
- An FA that accepts ab*a defined over Σ={a,b}.



- Transition Table.



	a	b
0	*1*	err
1	2	*1*
2	err	err



```
int trans_table[NSTATES][NCHARS];
int accept_states[NSTATES];
int state = INITIAL;
while(state != err){
        c = input.read();
        if(c == EOF ) break;
        state=trans_table[state][c];
}
return accept_states[state];
```

- Options: Logic 1, Logic 2