

# 5 Automata

Monday, September 9, 2024 12:41 PM

## Mathematical languages

Problems  $\begin{cases} \rightarrow \text{Specification: RegEx} \\ \rightarrow \text{Recognition: Automata} \end{cases}$

$$L = (+|-)? [0-9]^+ (\cdot [0-9]^+)? 'e' (+|-)? [0-9]^+$$

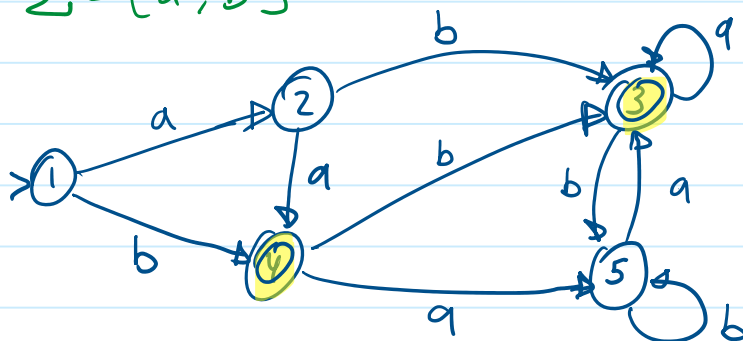
$27.3 + e 2.7 \in L ?$  No.

- Automata  
a mathematical Machine.

DEF:

- Set of states
  - initial state
  - subset of "accept" states.
- Transition function  
(state, symbol)  $\rightarrow$  state.

E.g.  $\Sigma = \{a, b\}$



How to recognize:

How to recognize:

Given a word  $w$

- start at the initial state and follow the transition function for each symbol in  $w$

e.g.  $a^1 b^2 a^3 b^3 a^5 b^5 a^3$

- If  $w$  ends in an accept state then  $w \in L$  otherwise  $w \notin L$

$L = \{ b, aa, ab, bb, aab, bba, aba, \dots \}$   
 not in  $L$   $L = \{ bab, a, abb, \dots \}$

**THEOREM:**

for every language  $L$  specified by a RegEx there exists an automata that recognizes  $L$

E.g.

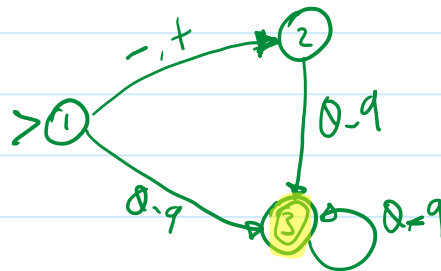
Integer Literal

$(+|-)? [0-9]^+$

Automata

$\Sigma = \text{ASCII}$

$\emptyset$        $-77$  ✓  
 $+82$        $-0001$



$12+3$  ✗  
 $\begin{matrix} \uparrow \uparrow \uparrow \\ 3 \ 3 \end{matrix}$

E.g. Scientific Numbers

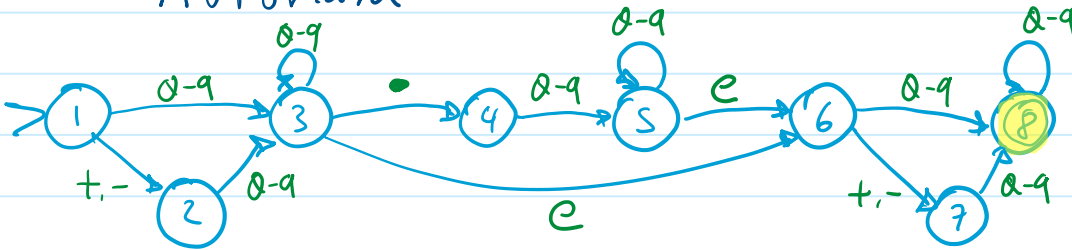
$(+|-)? [0-9]^+ (\cdot [0-9]^+)? 'e' (+|-)? [0-9]^+$

Automata

$a-a$

$a-a$

# Automata



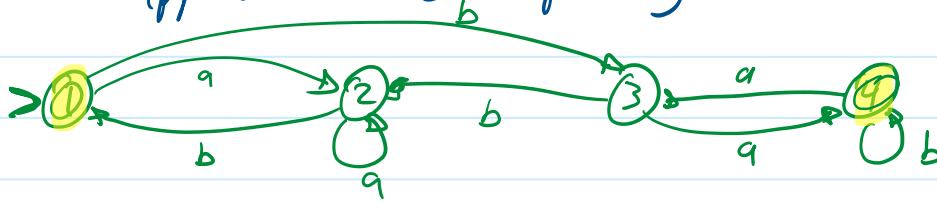
## • Encoding an Automata.

(write an automata in code)

- Loop over a "switch" statement
- Every 'case' of the switch statement corresponds to a state in the automata.
- the body of each case, look the current symbol and applies the corresponding transition function.

E.g.

$\Sigma = \{a, b\}$



$L = \{ \epsilon, ab, ba, bab, aab, \dots \}$

Pseudocode.

```
FUNCTION recognize ( string w ) : BOOLEAN
```

```
VAR state, i : INTEGER
```

```
    c : CHAR
```

```
    acc : BOOLEAN
```

```
BEGIN
```

```
    state := 1; i:=0; acc := False;
```

```
    WHILE i < len(w) DO
```

```
        c := w[i]
```

```
        CASE state OF
```

```
            1 : IF c = 'a' THEN state := 2
```

```
                IF c = 'b' THEN state := 3
```

```
            2 : IF c = 'a' THEN state := 2
```

```
                IF c = 'b' THEN state := 1
```

```
            3 : IF c = 'a' THEN state := 4
```

```
                IF c = 'b' THEN state := 2
```

```
            4 : IF c = 'a' THEN state := 3
```

$w = aba$   
 $\underbrace{\quad\quad\quad}_c$

state = 1 2 2 2

```
        IF c = 'b' THEN state := 4
    END
    i := i + 1
END
IF state = 1 OR state = 4 THEN
    acc = True
END
return acc
END.
```

We could also:

- Add fail/sink states.
- Return early
- Turn switch statement into lookup table.

	a	b
1	2	3
2	2	1
3	4	2
4	3	4

- other simplifications.

This is HW #1!!

— 0 — EOF