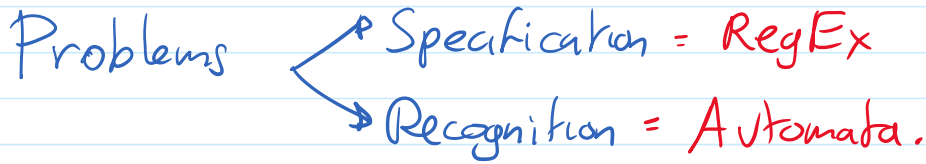


Mathematical Languages

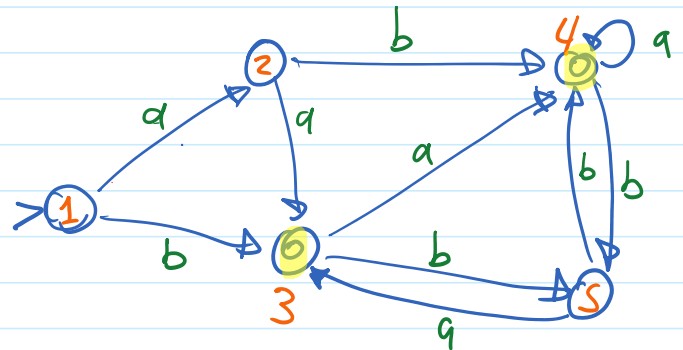
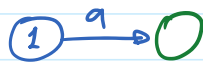


- Automata. a mathematical machine.

- DEF.
- Set of states
 - initial state
 - set of "accept" states
 - Transition function. (state, symbol from alphabet) \rightarrow state.

E.g

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



How to recognize?

- Given a word w start at the initial state and follow the transition function, for each symbol in w in order.

e.g. $abbaba$
 $\sim 2 \ 4 \ 3 \ 5 \ 3$

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

$$L = \{ \cancel{a}, b, aa, ab, \cancel{bb}, ba, aba, baa, \underset{hhh}{aaa}, \dots \}$$

$L = \{ a, b, aa, ab, \cancel{bb}, ba, aba, baa, aaa, \dots \}$
5
bbb.

THEOREM:

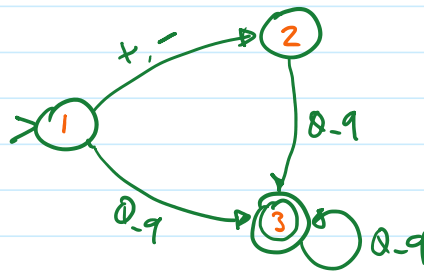
for every Language L specified by a RegEX
 There exists an automata that recognizes L

E.g.

Integer Constant

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

$\Sigma = \text{ASCII}$



0 -7

+82 -000

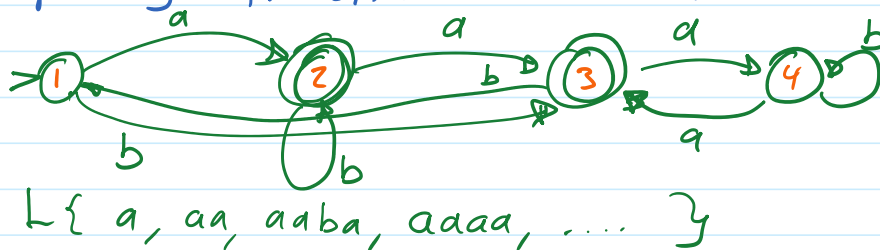
NOTE: - There is an algorithm that takes a regular expression and produces an automata

• Encoding an Automata:

(write an automata as code)

- Loop over a "switch" statement.
- Every case of the switch statements corresponds to a state in the automata.
- the body of each case should apply the corresponding transition function.

E.g:



Pseudocode:

FUNCTION recognize (string s) : BOOLEAN
 VAR state, i : INTEGER

```

    c : CHAR
    acc : BOOLEAN
BEGIN
    state := 1; i := 0; acc := false;

    WHILE i < length(s) DO
        c := s[i]
        CASE state OF
            1 : IF c = 'a' THEN state := 2
                ELSE state := 3
            2 : IF c = 'a' THEN state := 3
                ELSE state := 2
            3 : IF c = 'a' THEN state := 4
                ELSE state := 1
            4 : IF c = 'a' THEN state := 3
                ELSE state := 4
        END
        i := i + 1
    END
    IF state = 2 OR state = 3 THEN
        acc := true
    END
    RETURN acc
END.

```

We could:

- Add Fail/Sink states
- return Early
- Turn switch statement into lookup-table.
- Simplify when possible.

• This Algorithm we will use for HW #1!!