Alphabet … finite (unempty) set of symbols
|A| … size of alphabet A

Examples:

- A = \{ 'A', 'D', 'G', 'O', 'U'\}, |A| = 5
- A = \{0,1\}, |A| = 2
- A = \{\bigcirc, \bigcirc, \bigtriangleup\}, |A| = 3

Word (over alphabet A) … finite (maybe empty) sequence
also string of symbols of alphabet (A)
|w| … length of word w

Examples:

- w = OUAGADOUUGOU, |w| = 11
- w = 1001, |w| = 4
- w = □△○△□, |w| = 5
Language ... set of words (=strings) (not necessarily finite, can be empty)  
|L| ... cardinality of language L

1) Language specification -- List of all words in the language (only for finite language!)

Examples:

A₁ = \{'A', 'D', 'G', 'O', 'U'\}
L₁ = \{ADA, DOG, GOUDA, D, GAG\},  |L₁| = 5

A₂ = \{0, 1\}
L₂ = \{0, 1, 00, 01, 10, 11\},  |L₂| = 6

A₃ = \{□, □, △\}
L₃ = \{△△, ○□○, □□△○\},  |L₃| = 3
Language specification -- Informal (but unambiguous)
description in natural human language
(usually for infinite language)

Examples:


L₁: Set of all words over A₁, which begin with DA,
end with G a and do not contain subsequence AA.

L₁ = {DAG, DADG, DAGG, DAOG, DAUG, DADAG, DADDG... }  
|L₁| = ∞

A₂ = {0,1}

L₂: Set of all words over A₂, which contain more 1s than 0s
and where each 0 is followed by at least two 1s.

L₂ = { 1, 11, 011, 0111, 1011, 1111, ... , 011011, 011111, ... }  
|L₂| = ∞
Finite automaton

is a five-tuple \((A, Q, \sigma, S_0, Q_F)\), where:

- **A** ... alphabet ... finite set of symbols
- \(|A|\) ... size of alphabet
- **Q** ... set of states (often numbered)  (what is „a state“ ?)
- \(\sigma\) ... transition function ... \(\sigma : Q \times A \rightarrow Q\)
- **S_0** ... start state \(S_0 \in Q\)
- \(Q_F\) ... unempty set of final states \(\emptyset \neq Q_F \subseteq Q\)
Automaton FA1:

A \quad \ldots \text{alphabet} \ldots \{0,1\}, \quad |A| = 2

Q \quad \ldots \text{set of states} \{S, A, B, C, D\}

\sigma \quad \ldots \text{transition function} \ldots \sigma: Q \times A \to Q : \{
\sigma(S,0) = S, \quad \sigma(A,0) = B, \quad \sigma(B,0) = C, \quad \sigma(C,0) = C, \quad \sigma(D,0) = D,
\sigma(S,1) = A, \quad \sigma(A,1) = D, \quad \sigma(B,1) = D, \quad \sigma(C,1) = A, \quad \sigma(D,1) = D \}

S_0 \quad \ldots \text{start state} \quad S \in Q

Q_F \quad \ldots \text{unempty set of final states} \quad \emptyset \neq \{ C \} \subseteq Q

Transition diagram of the automaton FA1
Finite Automata

Example 6
Finite Automata

Example 7
Finite Automata

Example 8
When the last word symbol is read, automaton FA1 is in final state.

⇒

Word 01000100 is accepted by automaton FA1.
Finite Automata

Example 10
Finite Automata

Example 11
Finite Automata

Example 12
When the last word symbol is read, automaton FA1 is in a state which is not final. 

⇒

Word $\text{1 0 0 1}$ is not accepted by automaton FA1.
Finite Automata

Example 14
Finite Automata

Example 15

FA1

Finite Automata

Example 15

FA1
No word starting with $11\ldots$ is accepted by automaton FA1
No word containing $\ldots\ 11\ldots$ is accepted by automaton FA1
No word containing $\ldots\ 101\ldots$ is accepted by automaton FA1

Automaton FA1 accepts only words -- containing at least one 1
-- containing at least two 0s after each 1

Language accepted by automaton $X = \text{set of all words accepted by } X$
Automaton A activity:

At the beginning, A is in the start state.

Next, A reads the input word symbol by symbol and transits to other states according to its transition function.

When the word is read completely A is again in some state.

If A is in a final state, we say that A accepts the word,

if A is not in a final state, we say that A does not accept the word.

All words accepted by A represent a language accepted (or recognized) by A.
Language over alphabet \{0,1\}:

If a word starts with 0, it ends with 1,
If a word starts with 1, it ends with 0.

Example of analysis of different words by FA2:

0 1 0 1 0 :  (S),0 \rightarrow (A),1 \rightarrow (B),0 \rightarrow (A),1 \rightarrow (B),0 \rightarrow (A)
(A) is not a final state, word 0 1 0 1 0 is rejected by FA2.

1 0 1 1 0 :  (S),1 \rightarrow (C),0 \rightarrow (D),1 \rightarrow (C),1 \rightarrow (C),0 \rightarrow (D)
(D) is a final state, word 1 0 1 1 0 is accepted by FA2.
Language:
{ 0 1 0, 0 1 1 0, 0 1 1 1 0, 0 1 1 1 1 0, ...

Example of analysis of different words by FA3:

0 1 0 1 0 :  (S),0 → (A),1 → (B),0 → (C),1 → (D),0 → (D)
(D) is not a final state, word 0 1 0 1 0 is rejected by FA3.

0 1 1 1 0 :  (S),0 → (A),1 → (B),1 → (B),1 → (B),0 → (C)
(C) is a final state, word 0 1 1 1 0 is accepted by FA3.
Automaton FA4 accepts each word over the alphabet \{0,1\} which contains subsequence \[ ... \ 0 \ 1 \ 0 \ ...
\]

Example of analysis of different words by FA4:

0 0 1 0 1 : \( (S), 0 \rightarrow (A), 0 \rightarrow (A), 1 \rightarrow (B), 0 \rightarrow (C), 1 \rightarrow (C) \)

\( (C) \) is a final state, word **0 0 1 0 1** is accepted by FA4.

0 1 1 1 0 : \( (S), 0 \rightarrow (A), 1 \rightarrow (B), 1 \rightarrow (S), 1 \rightarrow (S), 0 \rightarrow (A) \)

\( (A) \) is not a final state, word **0 1 1 1 0** is rejected by FA4.
Language over the alphabet \{ +, - , . , 0, 1, ..., 8, 9, ... \} whose words represent decimal numbers

Example of word analysis

+87.09: \( (0),+ \rightarrow (1),8 \rightarrow (2),7 \rightarrow (2),. \rightarrow (3),0 \rightarrow (4),9 \rightarrow (4) \)

(4) is a final state, word +87.05 is accepted by FA5.

76+2: \( (0),7 \rightarrow (2),6 \rightarrow (2),+ \rightarrow (5),2 \rightarrow (5) \)

(5) is not a final state, word 76+2 is not accepted by FA5.
int isDecimal(char arr[], int length) {
    int i;
    int state = 0;

    for(i = 0; i < length; i++) {  // check each symbol
        switch (state) {
        ...

Code of the finite automaton

(The word which is being read is stored in the array arr[ ]):
case 0:
    if ((arr[i] == '+') || (arr[i] == '-')) state = 1;
    else if ((arr[i] >= '0') && (arr[i] <= '9')) state = 2;
    else state = 5;
    break;
case 1:
    if ((arr[i] >= '0') && (arr[i] <= '9')) state = 2;
    else state = 5;
    break;
case 2:
  if ((arr[i] >= '0') && (arr[i] <= '9')) state = 2;
  else if (arr[i] == '.') state = 3;
  else state = 5;
  break;
```c
3

```
if ((state == 2) || (state == 4)) // final states!!
    return 1;                    // success - decimal OK
else
    return 0;                    // not a decimal

} // end of function isDecimal()