Two Faces of Computation

- **Function Computation** : given an argument x compute the value of $f(x)$, where f is a function.
- **Language Membership** : given a string x and a language L, decide whether $x \in L$.

Function Computation : Examples

- Computation of factorial function.
- Sorting a sequence of integers.
- Reversing a string of characters.

Factorial Function

- We compute the factorial function from the set of natural numbers to itself ($\mathbb{N} \longrightarrow \mathbb{N}$).
- Graph of '!' as a language :
  - $graph(!) = \{(0,1),(1,1),(2,2),(3,6),(4,24),\cdots\}$.
  - Let the *alphabet* be $\Sigma = \{0,1,2,\cdots,9,*\}$.
  - Encoding of *graph*(!) and its language: $L_! = \{0*1, 1*1, 2*2, 3*6, 4*24, 5*120, \cdots\} \subseteq \Sigma^*$.
  - $6*720 \in L_!$, but $4*29 \notin L_!$.

Sorting of Integers

- Sorting of a finite sequence of integers may be viewed as a function from $\mathbb{N}^*$ **to itself**.

$$sort(40, 20, 30, 10, 5, 25) = (5, 10, 20, 25, 30, 40).$$

- This also has a counterpart in language.

Decision Problem and Language Membership

- **Test for primality** - whether a positive integer n is **prime**.
- **Test for syntatic correctness** : whether a C program is *well-formed*.
- **Test for reachability** - whether there is a path from a node s to another node d in a undirected graph G.
- **Existence of solution** - whether an equation over integers has an integral solution.

Every Decision Problem Can be Encoded as Language Membership Problem

Characterisic Function

Let the domain of our interest be D.

- Let $B \subseteq D$, the **characteristic function** of B with respect to D is a function μ_B from $D \longrightarrow \{0, 1\}$, so that

$$\mu_B(d) = \begin{cases} 1 & \text{if } d \in D, \\ 0 & \text{otherwise.} \end{cases}$$

Testing Prime

- **Input** : an element of $\mathbb{N}$.
- **Output** : *true* or *false*.
- The set **prime** $= \{2, 3, 5, 7, 11, 13, 17, \cdots\}$ and the set **nonPrime** $= \{0, 1, 4, 6, 8, 9, 10, 12, 14, \cdots\}$.
- The testing algorithm actually computes the *characteristic function* of **prime**.
- Test for **primality** may be viewed as a decision problem of the language **prime**.

Syntactic Correctness of C Program

- Let the **alphabet** of C language be Σ_C.
- Each C program is a syntactically correct string over Σ_C.
- Syntax checking by a C compiler is testing whether a given string belongs to C language.

Graph

- An **undirected graph** $G = (V, E)$ is a pair of data. The finite set V of **vertices** and the set $E = \{\{u, v\} : u, v \in V\}$ of **edges**. An undirected **edge** $\{u, v\} \in E$ may be written as $u - v$.
- A **directed graph** $G = (V, E)$ is also a pair of data. The finite set V of **vertices** and the set $E = \{(u, v) : u, v \in V\}$ of **edges**. An directed **edge** $(u, v) \in E$ is also written as $u \rightarrow$.

A Graph as a String

- A *graph* may be represented as a string over the alphabet $\Sigma = \{0, \cdots, 9, ;, (,), -\}$.
- Here efficiency is not our consideration!

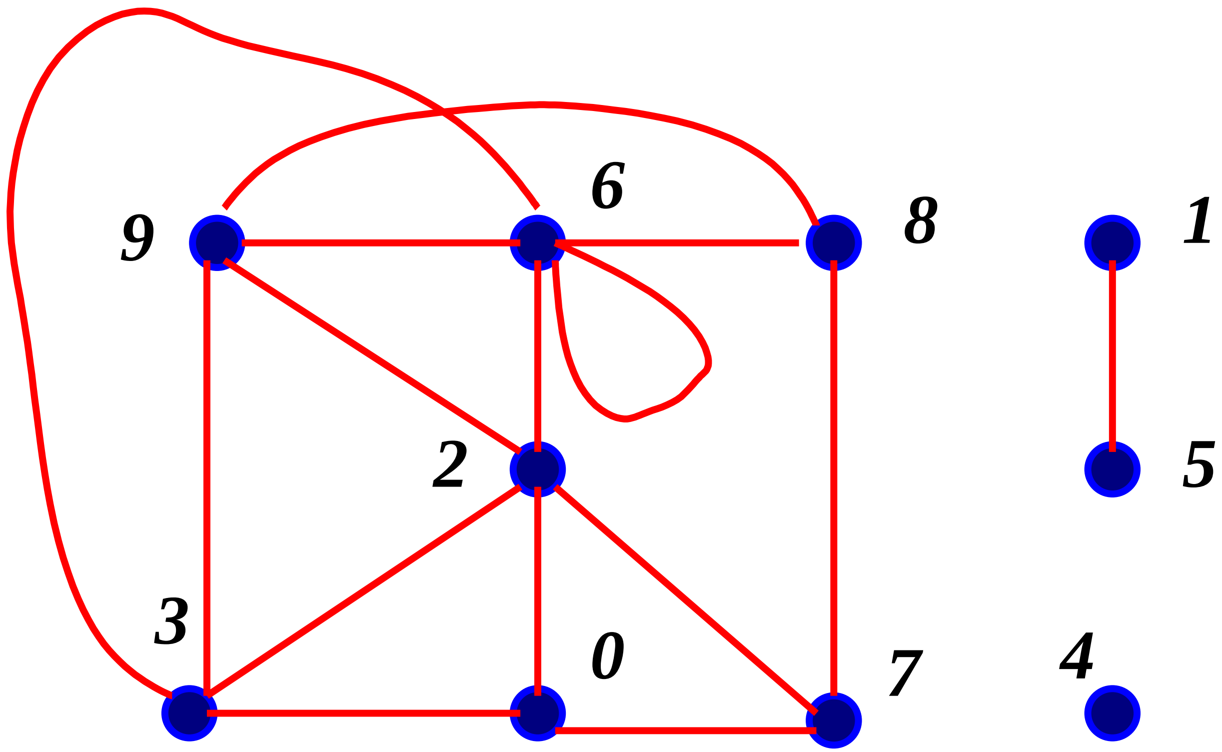

Figure 1: **Undirected Graph**

Graph Representation : An Example

- $V = \{0, 1, \cdots, 9\}$.
- $E = \{0-2, 0-3, 0-7, 1-5, 2-3, 2-6, 2-7, 2-9, 3-6, 3-9, 6-6, 6-8, 6-9, 7-8, 8-9\}$.
- The representation is
  $(0; 1; 2; 3; 4; 5; 6; 7; 8; 9); (0-2; 0-3; 0-7; 1-5; 2-3; 2-6; 2-7; 2-9; 3-6; 3-9; 6-6; 6-8; 6-9; 7-8; 8-9)$
- The following one is not a valid representation.
  $(0; 1; 2); (1-2-3;)$

Vertex Reachable Problem

- Given an undirected graph G and a pair of vertices s and d, whether the vertex d is reachable from the vertex s.
- We define a language **L** over the alphabet $\Sigma = \{0, \cdots, 9, ;, (,), -\}$ so that a string of the form

$$(s; d); (v_0; \cdots; v_{k-1}); (e_0; \cdots; e_{n-1}) \in L$$

where $V = \{v_0, \cdots, v_{k-1}\}$, $E = \{e_0, \cdots, e_{n-1}\}$, $s, d \in V$ and there is a **path** from the vertex s to the vertex d.

Vertex Reachable Problem (*cont.*)

- Any **vertex reachable** problem may be viewed as a decision problem of the language L.

Integer Solution of an Equation

- Consider a **univariate polynomial equation**

$$a_n x^n + \cdots + a_1 x + a_0 = 0,$$

  where the coefficients a_is are integers. The question is whether the equation has an **integral root**.

- This decision problem can be encoded as a decision problem of a language.

Study of Language Classes

- In our discussion we tried to show that **decision problems** of any **representable domain** can be **easily translated** to a **decision problem** of some **formal language**.
- The study of **language theoretic decision problems** is an **abstraction** of the study of decision problems in different domains.