## Introduction to Finite Automata

Languages
Deterministic Finite Automata
Representations of Automata

#### Programs as functions

- We consider the task of writing a program which performs a task as computing a function.
- Example: Suppose we want to write a program which calculates gcd of two numbers.
  - We want to compute the function  $gcd: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ .

#### Programs as functions

- Example 2: Suppose we want to write a program which sorts an array of length 10.
  - We are computing the function  $sort: \mathbb{N}^{10} \to \mathbb{N}^{10}$

#### **Decision Problems**

- A decision problem is a function with a one-bit output: "yes" or "no".
- Examples
  - IsPrime(n): a function which determines whether input number n is a prime number
  - IsConnected(G): a function which determines input graph G is connected
- Question: Can any general function be expressed as a decision problem?

#### Question

- Can any general function be expressed as a decision problem?
- Yes!
  - Consider  $f: A \rightarrow B$
  - Its 'decision problem version' is  $f_d: A \times B \rightarrow \{Yes, No\}$
  - $\int_{a}^{b} f(a,b) = \int_{a}^{b} Yes \quad \text{if } f(a) = b$   $No \quad \text{otherwise}$

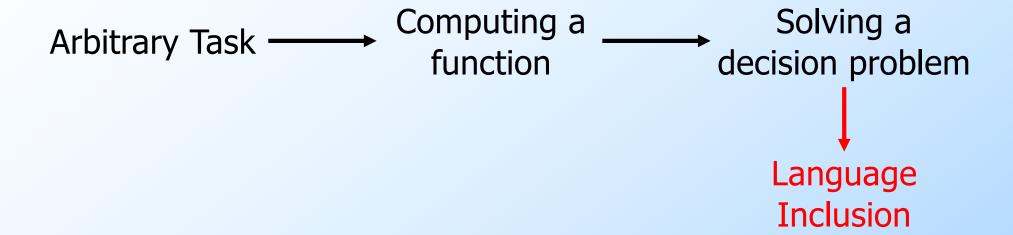
# Decision Problems: Alternative Specification

- ◆A Decision problem f can be specified by
  - the set A of all possible inputs
  - the subset  $B \subseteq A$  of "yes" instances
- Examples:
  - For the IsPrime(n) function,  $A = \mathbb{N}$ ,  $B = Set \ of \ all \ prime \ numbers = \mathbb{P}$
  - For the IsConnected(G) function, A = Set of all graphs, B = Set of all connected graphs

## Decision Problems as Language Inclusion

- Given  $x \in A$ , determining whether f(x) = Yes is equivalent to determining whether  $x \in R$
- Any decision problem can be expressed as a language inclusion problem! sen sentence is valid according to the rules (grammar) of the language.
  - A=Set of all sentences, B=Set of valid sentences.

## From Computation to Languages



## Language Inclusion: Abstraction

- We will always consider the input space to be the set of finite-length strings over a fixed, finite alphabet.
- ◆For IsPrime(n), the input space is the set of natural numbers.
  - Each natural number can be seen as a string over alphabet {0,1,2, ..., 9}.
- In general, any input can be encoded as a string.

#### **Alphabets**

- An *alphabet* is any finite set of symbols.
- ◆ Examples: English Alphabet, ASCII, {0,1,2,...,9} (*decimal alphabet*), {0,1} (*binary alphabet*).
- •We will use the Greek letter Σ to denote an alphabet.
  - Elements of an alphabet will be denoted by a, b, c, ...

#### Strings

- $\bullet$  A *string* over Σ is any finite-length sequence of elements of Σ.
- Examples:
  - For  $\Sigma = \{a,b\}$ , ab,ba,aba,aa,abaab are all distinct strings.
- We will use x, y, z, ... to denote strings.

### String length

- Length of a string is the number of symbols in the string.
  - For  $\Sigma = \{0,1\}$ , *011011* is string of length 6.
- ◆There is a unique string of length 0, denoted by ∈
  - Also called the *empty string* or *null string*

#### String Power Notation

- For  $a \in \Sigma$ , we write  $a^n$  for a string of a's repeated n times.
  - $a^5 = aaaaa, a^1 = a, a^0 = \epsilon$
- The set of all strings over alphabet Σ is denoted by  $Σ^*$ .

  - $\{0,1,2,...,9\}^* = ???$
  - $\{0,1,2,...,9\}^* = \{\epsilon,0,00,001,...\} \neq \mathbb{N}.$
- $\bullet$  By convention, we define  $\emptyset^* = \{\epsilon\}$

#### String Concatenation

- Concatenation takes two strings x and y and makes a new string xy by putting them together.
  - $x\epsilon = x$
- $\bullet x^n$  denotes the string obtained by concatenating n copies of x.
  - $x^0 = \epsilon$
  - $x^{n+1} = x^n x$

### String Concatenation

#### Examples

- $(ab)^5 = ababababab$
- $(ab)^1 = ab$
- $(ab)^0 = \epsilon$
- $(12)^2 = ???$
- $(12)^2 = 1212$

#### Set Concatenation

- We will denote sets of strings (subsets of  $\Sigma^*$ ) by symbols such as A, B, C, ...
  - Also called *languages*.
- •Given two sets A, B, the set concatenation AB is defined to be the set  $\{xy \mid x \in A \text{ and } y \in B\}$ 
  - Example:  $\{a,b\}\{ab,ba\} = \{aab,aba,bab,bba\}$