INTRODUCTION TO DATA STRUCTURES

iACADEMY
SCHOOL OF COMPUTING • SCHOOL OF BUSINESS • SCHOOL OF DESIGN

SCHOOL OF COMPUTING

BENNETT M. TANYAG

DEFINITION OF DATA STRUCTURE

- Computing systems are concerned with the storage and retrieval of information.

- For systems to be economical the data must be organized in such a way as to support efficient manipulation (by algorithms).

- A ***data structure*** is an arrangement of data for the purpose of being able to store and retrieve information.

- In other words, data structures allow a programmer to determine how to represent real world information/data in a program.

- Choosing the wrong data structures (and wrong algorithms) makes a program slow and unmaintainable.

DEFINITION OF DATA STRUCTURE

- Example: Sets

{25, 33, 12, 47, 14, 5, 75}

- Example: Polynomials

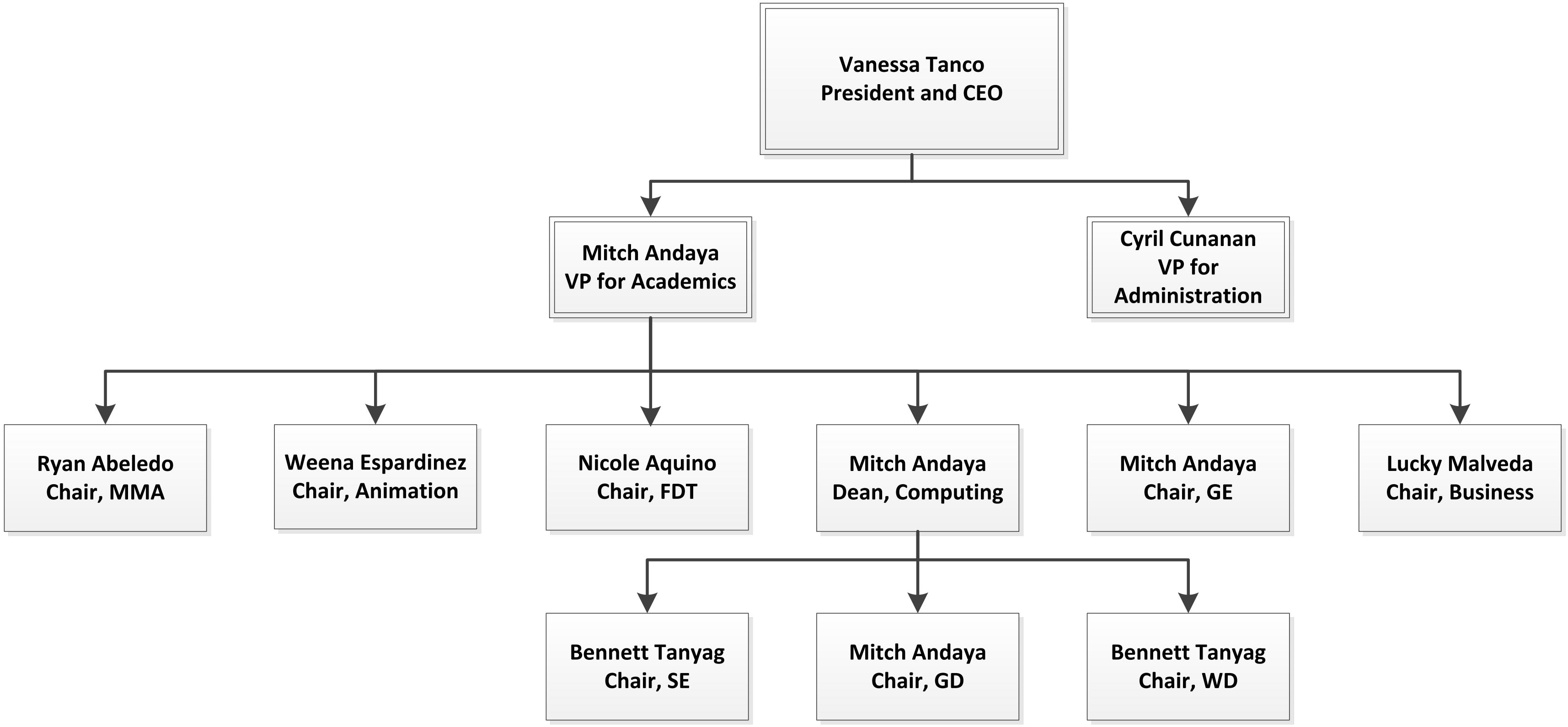
$$5x^5 - 15x^4 + 6x^3 + 12x^2 - 9x + 17$$

- Example: Matrices

$$\begin{bmatrix} 7 & 8 & 3 & 4 \\ 2 & 9 & 7 & 5 \\ 3 & 1 & 4 & 0 \\ 8 & 6 & 5 & 2 \end{bmatrix}$$

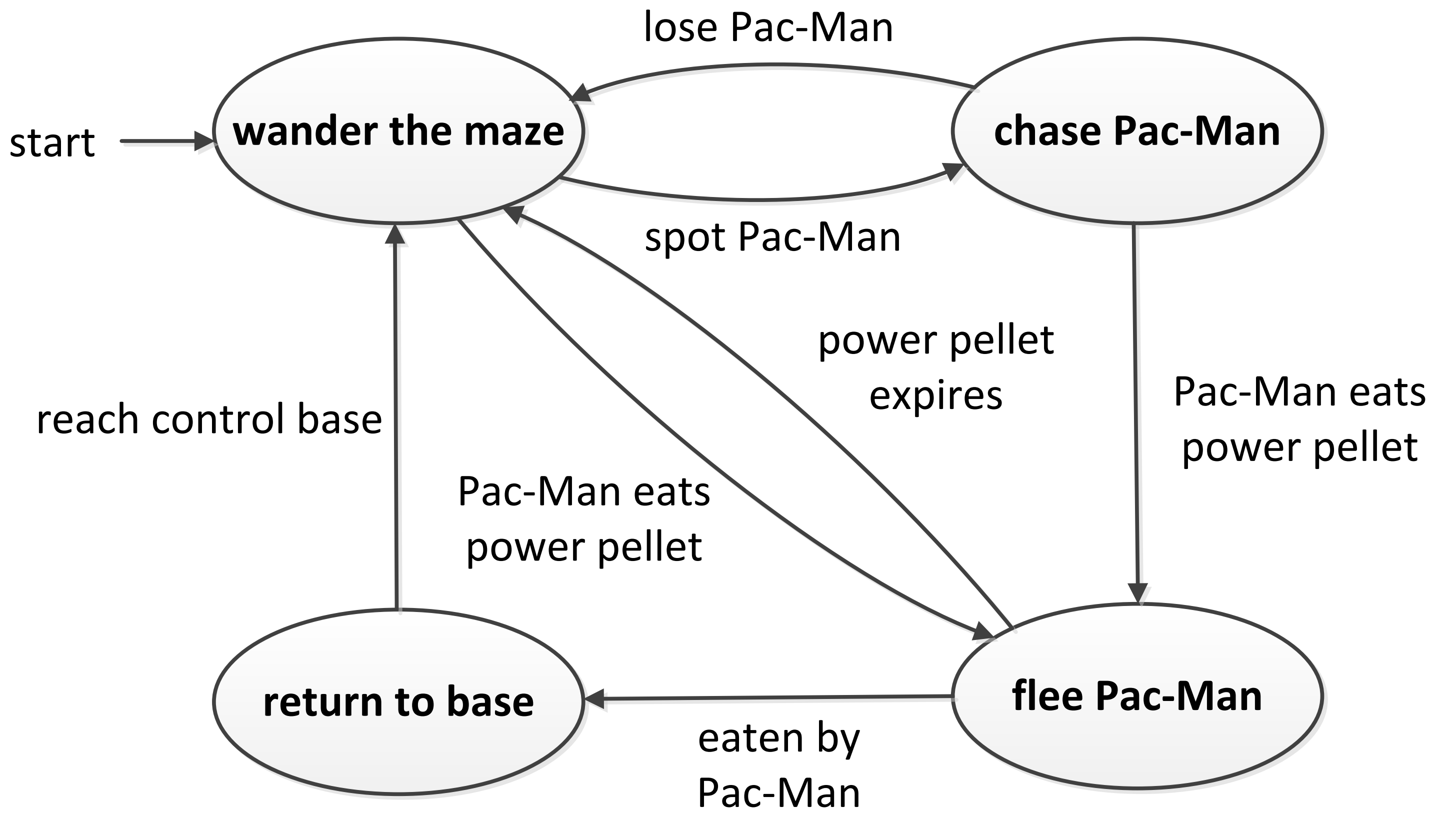
DEFINITION OF DATA STRUCTURE

- Example: Organizational Chart



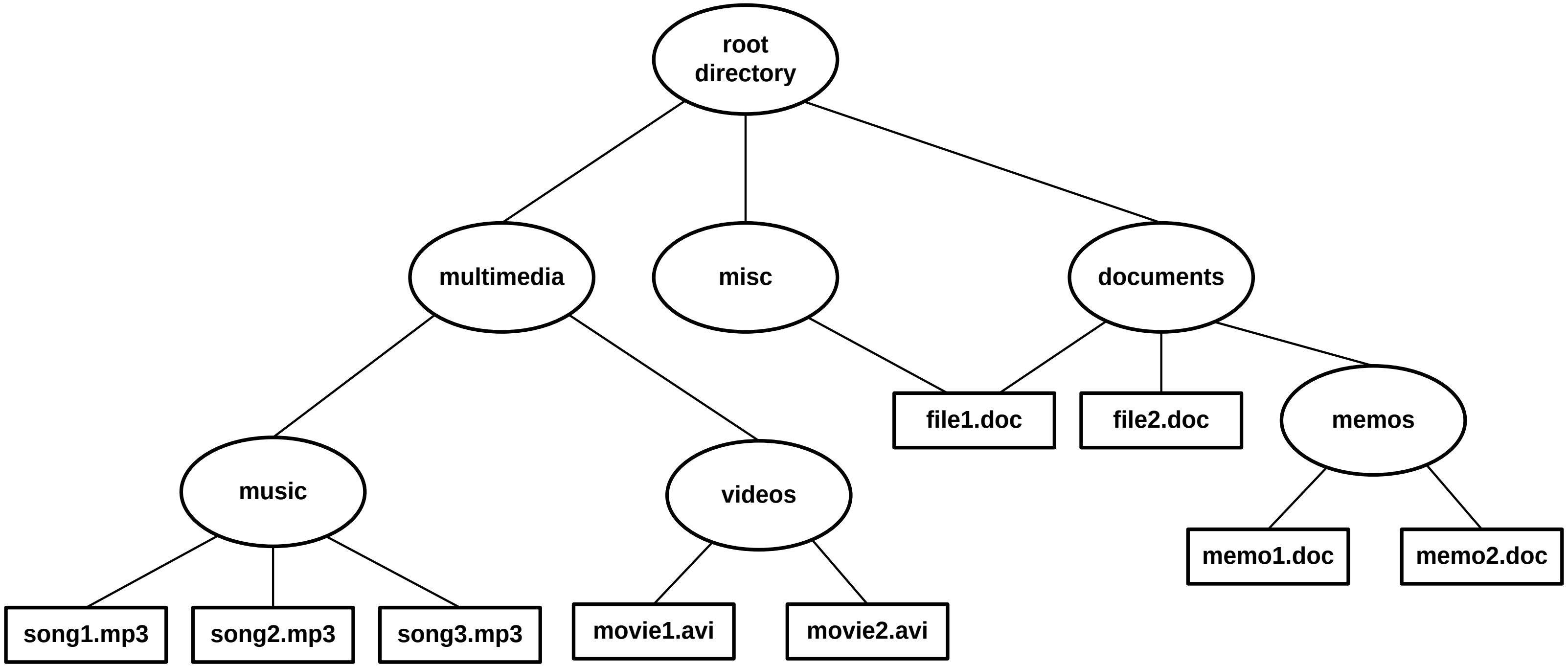
DEFINITION OF DATA STRUCTURE

- Example: Finite State Machine of the Behavior of a Ghost in Pac-Man



DEFINITION OF DATA STRUCTURE

- Example: Computer File System



ABSTRACT DATA TYPES

- A ***data type*** is a classification of the type of data that a variable can hold in computer programming.

- A data type is characterized by:
  - A set of values
  - A set of operations, which can be applied uniformly to all these values

- Basic or Primitive Data Types

Type	**Values**	**Operations**
Integer	… , -2, -1, 0, 1, 2, ….	+, -, *, /, %, ++, --, …
Floating Point	…. 0.0 ….	+, -, *, /, …
Character	'A', 'B', 'C', …	<, >, =, …

ABSTRACT DATA TYPES

- ***Abstract Data Types*** (ADT) are a mathematical specification of a set of data and the set of operations that can be performed on the data.

- Example: The Abstract Data Type Set

  – A ***set*** is a group of objects called *elements* (numbers, symbols, and even other sets) represented as a unit. The order of describing a set does not matter nor does repetition of its members.

  – Operations on Sets

    1. Add an element to a set
    2. Remove an element from a set
    3. Determine if an certain value/item is an element of a set
    4. Subtract one set from another
    5. Get the union of several sets
    6. Get the intersection of several sets
    7. Get the complement of a set

ABSTRACT DATA TYPES

- Example: The Abstract Data Type Matrix
  - A matrix is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns. The individual items in a matrix are called its *elements* or *entries*.
  - Operations on Matrices
    1. Assign a value at a designated row and column number
    2. Determine the value of the element at a designated row and column number
    3. Determine if a square matrix is a diagonal, upper triangular, lower triangular, identity, scalar, or a zero matrix
    4. Add, subtract, or multiply matrices
    5. Get the transpose of a matrix

ABSTRACT DATA TYPES

- Example: The Abstract Data Type Organizational Chart
  - An organizational chart a diagram that shows the structure of an organization and the relationships and relative ranks of its parts and positions/jobs.
  - Operations on Organizational Chart
    1. Change the title of a certain position
    2. Determine the name of the person holding a particular job title
    3. Change the name of the person holding a particular job title
    4. Determine the immediate superior of a particular person/title
    5. Determine the immediate subordinate(s) of a particular person/title
    6. Remove a certain job position from the chart
    7. Add a certain job position in the chart

ABSTRACT DATA TYPES

- ADTs are abstract in the sense that the focus is on the definitions and the various operations with their arguments.

- The actual implementation is not defined, and does not affect the use of the ADT.

- An ADT may be implemented by specific data types or data structures, in many ways and in many programming languages; or described in a formal specification language.