CSE 3010 – Data Structures & Algorithms

Lecture #6

What will be covered today

- Big O, Big Omega and Big Thetha
- Understanding abstract data type (ADT)
- Algebraic specifications to define ADTs

Notations for measuring complexity of algorithms

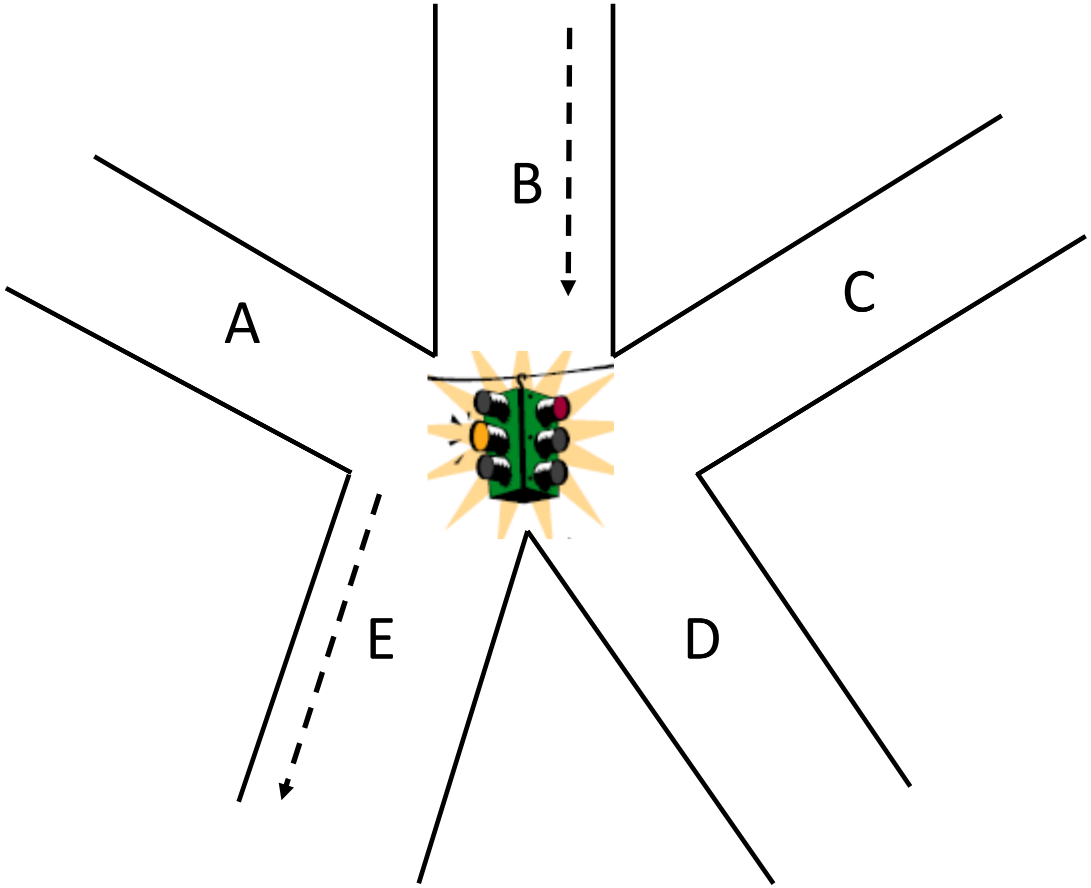
- Asymptotic notation
    - Performance of algorithms in relation to the input
- Big O notation
    - Gives upper bound
- Big-omega notation
    - Gives lower bound
- Big-theta notation
    - Gives both lower and upper bound

O notations do not give exact growth rate, they provide approximations, using = in place of ~

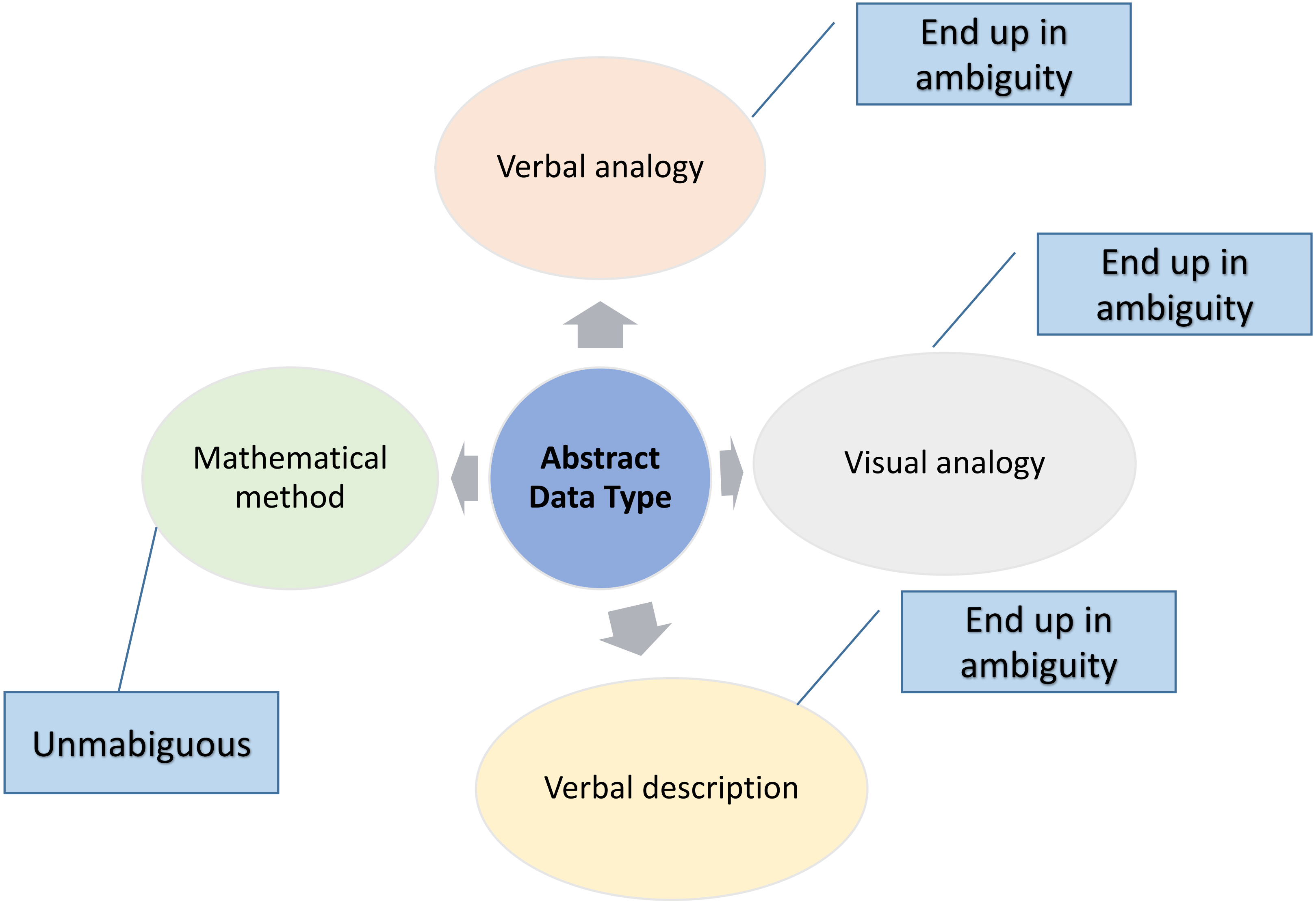
Abstract data types

- Data type
  - Set of values plus operations defined on them
  - Integer data type
    - Add, Multiply, Subtract, Divide, Modulus
  - Char data type
    - Predecessor, Successor
- Abstract data type
  1. Mathematical model with a collection of operations
  2. Specifies the logical properties of a data type
  3. Not concerned with space or time complexity
  4. Specifies only operations
  5. Implementation methods not specified
  6. Programming-language independent

Example



Ways to describe an ADT



Abstract Data Type

Verbal analogy

End up in ambiguity

Visual analogy

End up in ambiguity

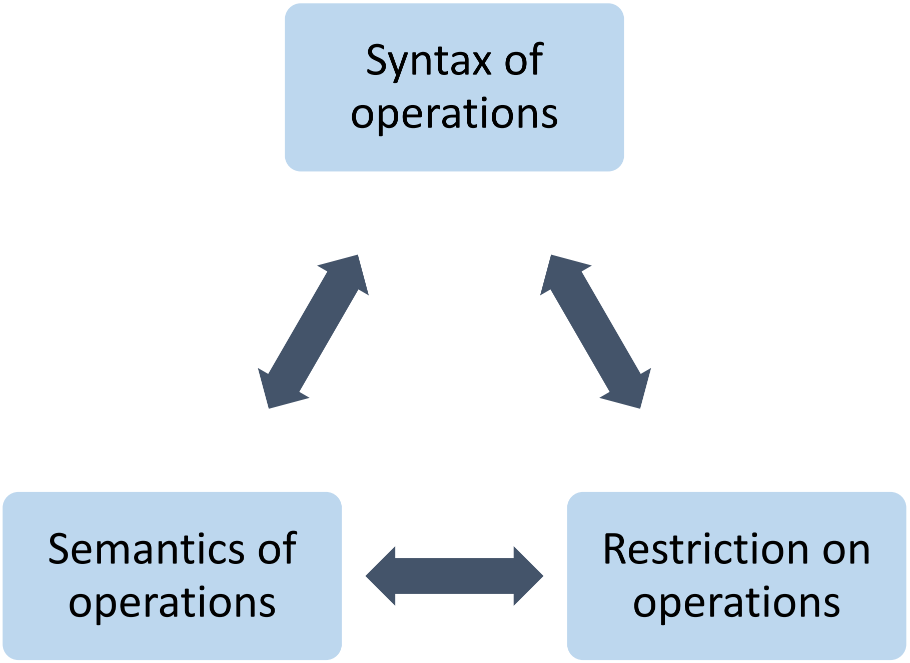
Verbal description

End up in ambiguity

Mathematical method

Unmabiguous

Using algebraic or axiomatic method to describe an ADT



Using type, operations and axioms for an ADT