Introduction

• Purpose
  ▪ Specification of external interfaces
    • Operations (arguments, results)

• Example
  ▪ Abstract data types
    • You define behavior of all the operations, and not the internal data representation
Algebraic method

- Using
  - Algebraic structures
  - Abstract data types

- ADT = carrier sets + operations + axioms
Basic theory
Algebra

- Algebra \( A = \langle D, F \rangle \)
  - Carrier set \( D \)
  - Functions \( F \)

- Function \( f_A \in F \)
  - \( f_A : A \times \ldots \times A \rightarrow A \)
  - \( f_A : \rightarrow A \)
Sorts

• Sort = data type
  ▪ Examples: Nat, Int, Bool, Strings, ...

• Many-sorted algebras

• Sub-sorting relation
  ▪ Nat < Int
Algebra - revisited

- **Notation**
  - $S$ ... sorts
  - $F$ ... functions (operations)
  - $D$ ... carrier sets (data)
  - $A$ ... algebra

- **Types of functions**
  - $T = S^* \times S$
  - $s_1 \times \ldots \times s_n \rightarrow s$

- **Algebra** $A = \langle [D_s]_{s \in S}, [F_t]_{t \in T} \rangle$
Example
Signature (S, Σ)

- Σ = [Σ_t]{t ∈ T}

Σ-algebra

- Carrier set D_s for every sort s ∈ S
- Operation f_A for each symbol f ∈ F
Properties of operations

- Basic approach
  - Equations between function expressions

- Set E of all equations (sentences, axioms)

- Executable specifications (models)
More complex signatures and equations

- Overloaded functions
  - Different subsorts
  - Number of arguments

- Predicates and relations
  - Signature: the set $P$ of predicate symbols
Initial model

- Exactly the right number of elements in carrier sets
  - “no junk and no confusion”

- Multiple isomorphic models
Algebraic specification

- Assumptions
  - Programs are modeled by many-sorted algebras
  - Correctness of the input/output behavior has precedence over all other properties

- $Q = (S, \Sigma, E)$

- Two parts
  - Declarations (signature)
  - Equations (semantics)
Example

- List of integers
  - Operations: add, remove, get, size, contains
    - insert and remove to/from any position

- Use of recursion

- Exceptions (errors)
Semantics of algebraic specifications

- $Q = (S, \Sigma, E)$
  - well-formed specification

- $\text{Sem}[Q]$
  - the class of all initial algebras (models)
Languages

- CASL: Common Algebraic Specification Language
  - http://www.informatik.uni-bremen.de/cofi/index.php/CASL
  - http://www.cofi.info

- Other: Larch (family), OBJ3, ASL
• Ian Sommerville: Software Engineering
  - consider just recent book editions (9th or 10th)