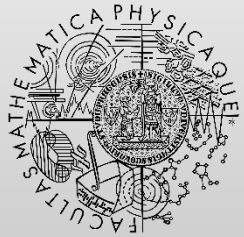
Algebraic Specification Methods & Languages

http://d3s.mff.cuni.cz

Pavel Parízek



FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Declarative specifications

- Concepts similar to functional programming
- Equations between inputs and desired outputs
- Hidden low-level implementation details

Functional programming concepts

- Procedures without side effects
  - Result is the new program state
- Pattern matching
  - case statements, functional call expressions
- Recursion instead of loops
- Tail recursion (algorithms)
- Algebraic data types

Introduction to algebraic specifications

- 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
- Usage: **prototyping**
  - Executable specifications

Algebraic method

- Using
  - Algebraic structures
  - Abstract data types

- ADT = carrier sets + operations + axioms

Basic theory

Algebra

- Algebra A = <D, F>
  - 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

- Signature $(S, \sum)$
  - $\sum = [\sum_t]_{\{t \in T\}}$
- $\sum$-algebra
  - Carrier set D_s for every sort $s \in S$
  - Operation f_A for each symbol $f \in F$

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 redundancy (“garbage”)
  - No ambiguity (“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
- $Q = (S, \sum, E)$
- 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
- Constructing bigger instances (values)
  - Lists with multiple elements
- Exceptions (errors)

Semantics of algebraic specifications

- Q = (S, ∑, E)
  - well-formed specification

- Sem[Q]
  - the class of all initial algebras (models)

Languages

- Maude
  - https://maude.cs.illinois.edu/
- CASL: Common Algebraic Specification Language
  - https://link.springer.com/book/10.1007/b11968
  - http://www.cofi.info
  - E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Bruckner, P.D. Mosses, D. Sannella, and A. Tarlecki. CASL: The Common Algebraic Specification Language. Theoretical Computer Science, 286(2), 2002
- Other: Larch (family), OBJ3, ASL

Realistic examples

- Maude
  - basic mutex
  - HTTP client
- CASL
  - steam-boiler control system

Literature

- Ian Sommerville: Software Engineering
  - consider just recent book editions (9th or 10th)
  - https://software-engineering-book.com/