Advanced Usage of Z: Objects & Refinement

Pavel Parízek
Object-Z

- Main features
  - Classes & instances
  - Operations (methods)
  - Inheritance
  - History invariants
  - Dot notation

- Benefits
  - OOP: structure, modularity, reuse
Refinement

- Goal: specification ➔ design ➔ code

- Operation refinement
- Data refinement
Operation refinement

- Abstract operation OpA
- Concrete operation OpC

- Weaker precondition
  - pre OpA => pre OpC

- Stronger postcondition
  - post OpC => post OpA

- Analogy: inheritance & method overriding
  - Object-oriented development
Data refinement

- Goal: design concrete data structures

- Abstract schemas $\implies$ abstract states
- Concrete schemas $\implies$ concrete states

- Abstraction schema: abstract $\leftrightarrow$ concrete

- Correct data refinement
  - $\text{pre \ OpA} \land \text{Abs} \implies \text{pre \ OpC}$
  - $\text{pre \ OpA} \land \text{Abs} \land \text{OpC} \implies \text{Abs'} \land \text{post \ OpA}$
  - $\text{InitC} \implies \text{InitA} \land \text{Abs}$
Iterative step-wise refinement

- Target: complex systems

- Step
  - Refine some parts of the system model
  - Create procedures ➔ modular design
Example

- Bank account system

- Abstract data structures
  - Mathematical model (clarity)

- Concrete data structures
  - Computer representation (performance)
• G. Smith. The Object-Z Specification Language
  - http://doi.org/10.1007/978-1-4615-5265-9