Advanced Usage of Z: Objects & Refinement

http://d3s.mff.cuni.cz

Pavel Parízek

Department of Distributed and Dependable Systems

FACULTY OF MATHEMATICS AND PHYSICS
Charles University
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 \( \text{OpA} \)
- Concrete operation \( \text{OpC} \)

- Weaker precondition
  - \( \text{pre OpA} \Rightarrow \text{pre OpC} \)

- Stronger postcondition
  - \( \text{post OpC} \Rightarrow \text{post OpA} \)

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

- Goal: design concrete data structures
- Abstract schemas \(\rightarrow\) abstract states
- Concrete schemas \(\rightarrow\) concrete states
- Abstraction schema: abstract \(\leftrightarrow\) concrete
- Correct data refinement
  - \(\text{pre OpA} \land \text{Abs} \Rightarrow \text{pre OpC}\)
  - \(\text{pre OpA} \land \text{Abs} \land \text{OpC} \Rightarrow \text{Abs'] \land post OpA}\)
  - \(\text{InitC} \Rightarrow \text{InitA} \land \text{Abs}\)
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