Covariant Conversions (CoCo): A Design Pattern for Type-Safe Modular Software Evolution in Object-Oriented Systems

George Heineman
(WPI)

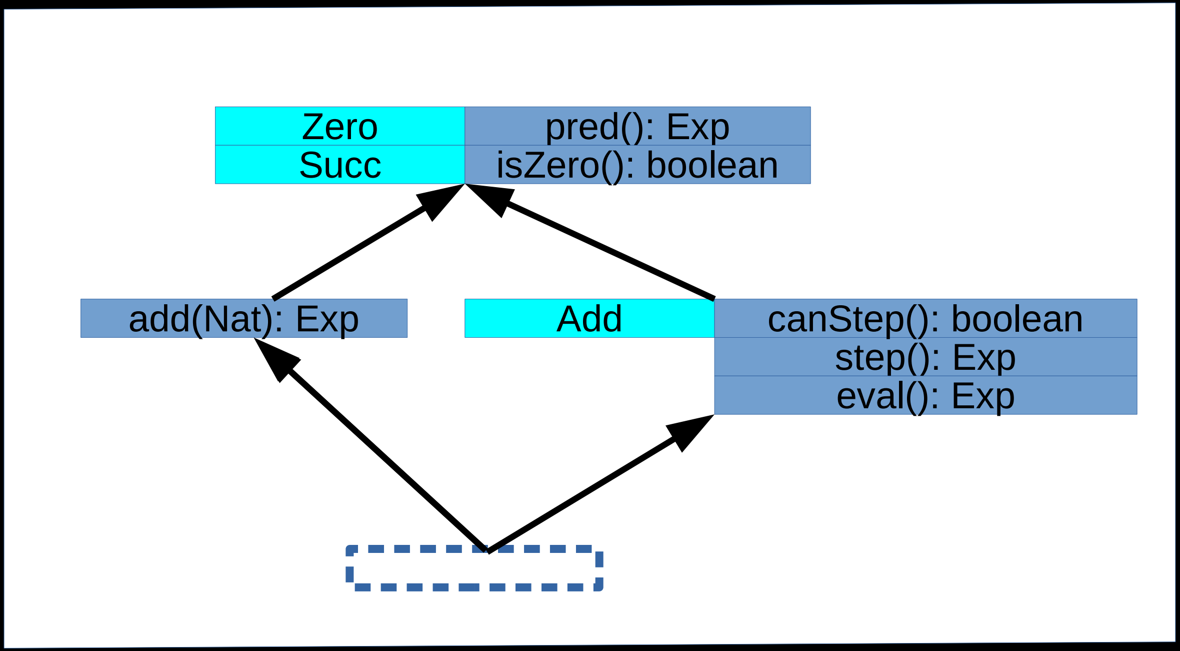
Boris Düdder
(DIKU)

Jan Bessai
(TU Dortmund)

CoCo: Covariant Conversions

- New *Design Pattern* to solve the Expression Problem
  - add methods & classes without casts, duplication, or modification of existing code
  - Old problem, 20 years of research, assumed to require advanced type systems
- Fully *Type-safe*
- Works *out of the box* in Java, Scala, C#, …
  - no language extensions or code-generators needed!
- Support for side-effects via setters
- Support for *producer and binary methods*
- Allows to *merge branching extensions*
- Acceptable amount of overhead

Example extension Graph



Zero

Succ

pred(): Exp

isZero(): boolean

add(Nat): Exp

Add

canStep(): boolean

step(): Exp

eval(): Exp

More in the Paper + Artifact

- More Examples (Java, Scala, C#)
  - XML Processing
  - Extended Standard EP Example
  - Compiler Suite from Types and Programming Languages
- Support for multiple subdomains
- Techniques for boilerplate reduction
- Benchmarks

Thanks!

source: wikimedia.org