ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Microsoft Research

M. Garcia,T. Gvero,H. Hojjat,V. Kuncak,M. Odersky,R. Piskac, L.Rytz

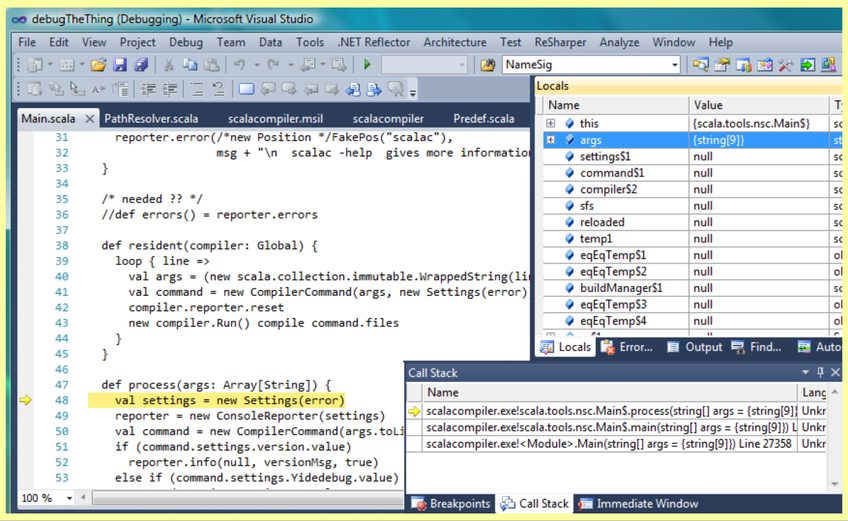
ProgLab.NET: A Workbench for Ensuring Software Quality and Reliability

- **ProgLab.Net:** programming environment equipped with a set of program analysis tools
- Improve the productivity of software development with rapid feedbacks from the system
- Increase the reliability of software by formal verification (sequential and concurrent)
- Aid the programmer by automatic generation of code snippets

.Net Translation

- Scala.NET targets all .NET platforms
  - desktop & server
  - mobile
  - game console
- Programs using only Scala SDK are cross-platform (JVM and .NET)
- Additionally, a migration tool (jdk2ikvm) ports JDK-based Scala sources to .NET

http://lamp.epfl.ch/~magarcia/ScalaNET/



Side-Effect Analysis

Methods declare their side-effects

A compiler-plugin checks the actual side-effects of the code

```
class Set[A] {
  def add(a: A): Unit @mod(this) = { ... }
  def map[B](f: A => B): Set[B] @pc(f.apply(_)) =
    { ... }
}

def singletonSet[A](a: A): Set[A] @mod() = {
  val s = new Set[A]()
  s.add(a)
  s}
```

Lightweight syntax for effect-polymorphism: the effect of `map` depends on the effect of `f.apply`

Non-observable side-effects can be masked

Eldarica: Predicate Abstraction for Concurrent Programs

Huge state space in a concurrent program
- unknown number of processes
- unbounded variables

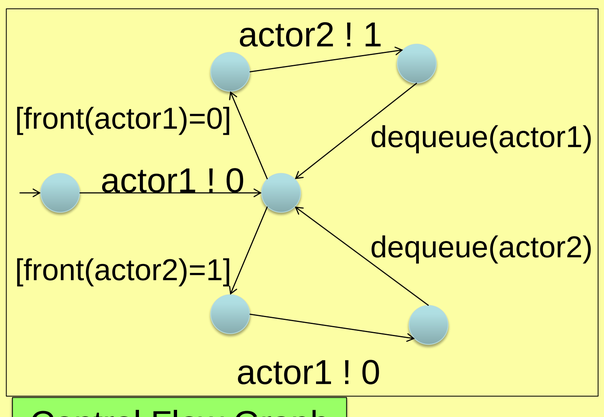
concrete space

Finite manageable state space
- locate error in abstract space
- generalize counter example in original program

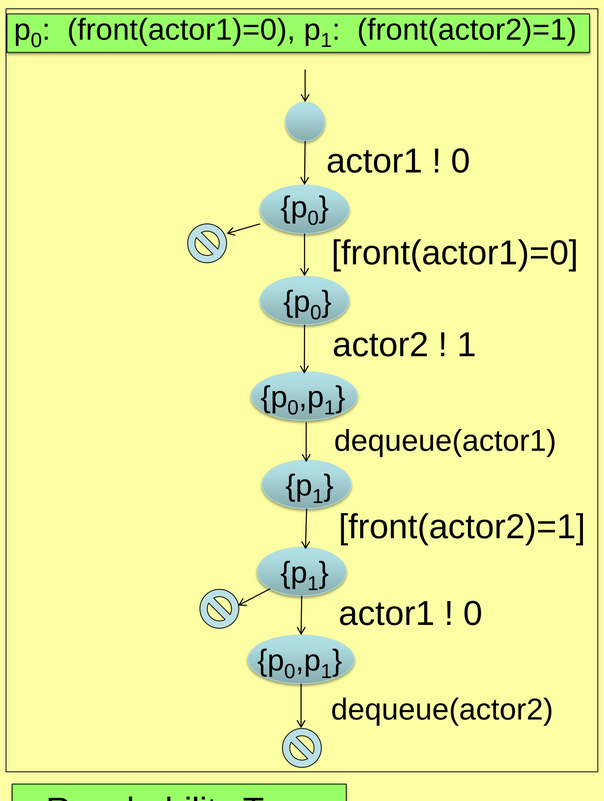
abstract space

Predicate Abstraction

Abstract with respect to 'important' features
(critical messages, shared variables,…)



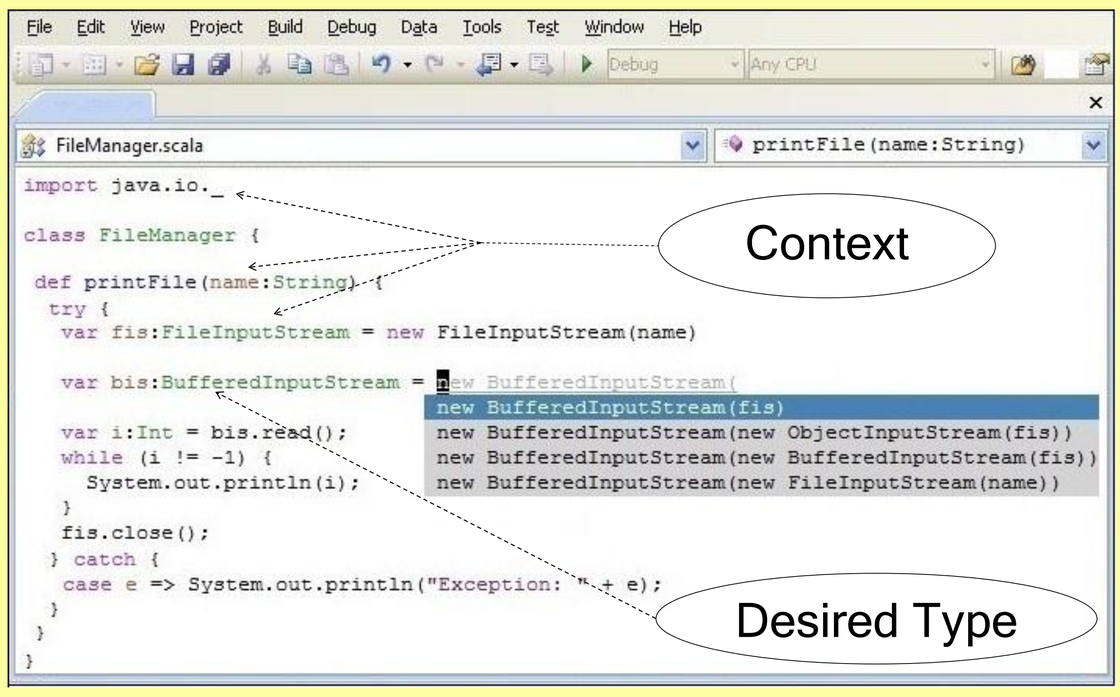
Control Flow Graph



Reachability Tree

- Reachability Tree: unfolding the control flow graph in an abstract domain
- Iterative process: refine the abstraction until:
  - detect genuine error
  - establish correctness

InSynth: Interactive Synthesis of Code



- Using Scala API classes and methods is **hard**
- Solution: Interactive Synthesis
  - Input: **desired type** and **declarations** visible in context
  - Encodes them to FOL formulas:
    - Desired type – goal
    - Declarations – axioms
  - Assigns them **weights**
  - Runs **resolution** algorithm modified to find solutions with smallest weights
  - Output: ranked expressions that have desired type – **code snippets**
- Supports: Generic types, higher order functions, sub-typing
- Uses machine learning
  - Uses source code **corpus to learn weights**
  - Higher the method frequency, smaller the weight
- Evaluation – 83 real-world API examples
  - Recreates expected snippet in 72 examples (**87%**)
  - Expected snippet has highest rank in 38 examples (**46%**)