

School of Computer and Communication Sciences



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE 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

S }

- Increase the reliability of software by formal verification (sequential and concurrent)
- Aid the programmer by automatic generation of code snippets

.Net Translation

Side-Effect Analysis

- 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/

😎 debugTheThing (Debugging) - Microsoft Visual Studio					
File Edit View Project Debug Team Data Tools .NET Reflector Architecture Test ReSharper Analyze Window Help					
1 m -	🛅 - 📂 - 💭 - 🖉 👔 🐇 👘 🛍	- 🏄 Name	eSig 🔹	🔩 🖀 🎲 🖄 🏷 🛃) <u>61</u>
		₩	Name	Value	T
Main.s	$_{ m cala}$ $ imes$ PathResolver.scala scalacompiler.msil scalac	ompiler Predef.scala	🗉 🥥 this	{scala.tools.nsc.Main\$}	sc
	31 reporter.error(/*new Position */FakePos(🗉 🖇 args	{string[9]}	st	
	32 msg + "\n scalac -help	gives more information	settings\$1	null	sc
	33 }		command\$1	null	sc
	34		compiler\$2	null	sc
	35 /* needed ?? */		🧳 sfs	null	so
	36 //del errors() = reporter.errors		💡 reloaded	null	so
	<pre>38 def resident(compiler: Global) {</pre>		💡 temp1	null	so
	39 loop { line =>		eqEqTemp\$1	null	ol
	40 val args = (new scala.collection.immut	able.WrappedString(li	eqEqTemp\$2	null	ol
	41 val command = new CompilerCommand(args	, new Settings(error)	buildManager\$1	null	sc
	42 compiler.reporter.reset		eqEqTemp\$3	null	ol
	43 new compiler.Run() compile command.fil	.es	eqEqTemp\$4	null	ol
4	44 }		🖂 Locals 📸 Error 🖃	Output 🗮 Find 🥅	Auto
4	45 }	procession and a second se			
	46	Call Stack		•	ų×
	4/ det process(args: Array[String]) { 48 val settings - new Settings(error)	Name		La	ing 🔶

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(_)) = { ... }

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

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



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

[front(actor1)=0]

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,...)

dequeue(actor1)

 p_0 : (front(actor1)=0), p_1 : (front(actor2)=1)

actor1 ! 0

InSynth: Interactive Synthesis of Code





actor2!1

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



- 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%)