Formal Verification

for Reallife Business

Golovach Ivan

RChain Coop

August, 2018

Contents

SYNTAX

What is Formal Verification?

Formal Verification

is the act of proving or disproving the correctness of algorithm with respect to a certain formal specification or property, using formal methods of mathematics.

.

Formal Specification (not Requirements):

- has a formal syntax
- semantics fall within one domain

Formal Verification VS QA/Testing

A Testing is checking **FEW** test cases.

.

A test case: (input, env) $\rightarrow$ $(output, env)$

$\sin(0^o) = 0$

$\sin(90^o) = 1$

$\sin(180^o) = 0$

$\sin(270^o) = -1$

.

.

A Formal Verification is checking spec/property in **ALL** cases.

.

Spec: $\sin(x) \equiv x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - x^{11}/11! + ...$

Prop: -1 $\leqslant$ $sin(x)$ $\leqslant$ 1

Specification

equation reasoning

Spec and impl use the **SAME** lang (usually)

.

int[] sort(int[] arr)

.

- Bubble sort
- Insertion sort
- Merge sort
- Quick sort
- Heap sort
- ...

Properties

Examples

Spec and impl use **DIFFERENT** langs (usually)

.

- Deadlock freedom
- Livelock freedom
- Eventual consistency
- Req/Resp correspondence
- Security: user data sandbox
- ...

.

Properties

Logics

- Modal $\mu - calculi$
- Hennessy-Milner Logic
- Separation Logic
- ...

Amazon way

Model Checking

Applying TLA+ to some of Amazon's more complex systems.

System	Components	Line Count (Excluding Comments)	Benefit
S3	Fault-tolerant, low-level network algorithm	804 PlusCal	Found two bugs, then others in proposed optimizations
	Background redistribution of data	645 PlusCal	Found one bug, then another in the first proposed fix
DynamoDB	Replication and group-membership system	939 TLA+	Found three bugs requiring traces of up to 35 steps
EBS	Volume management	102 PlusCal	Found three bugs
Internal distributed lock manager	Lock-free data structure	223 PlusCal	Improved confidence though failed to find a liveness bug, as liveness not checked
	Fault-tolerant replication-and-reconfiguration algorithm	318 TLA+	Found one bug and verified an aggressive optimization

Blockchain

Properties

- immutability (no updates/bugfixes)
- public code (hacker can read code)
- digital assets (directly steal money)

Formal Verification

Difficulties

You don't want:
- write formal specification (not requirements)
- know the tools (thoroughly)
- no Java/PHP/NodeJS/... (specific langs)
- no frameworks (side effects, uncontrolled functionality)