CM0859 – MT5009 Type Theory
Appendix: Testing with QuickCheck

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2025-2

Testing with QuickCheck

A paper

Koen Claessen and John Hughes (2000). QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. ICFP'00. DOI: https://doi.org/10.1145/357766.351266.

Testing with QuickCheck

A paper

Koen Claessen and John Hughes (2000). QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs. ICFP'00. DOI: https://doi.org/10.1145/357766.351266.

Most Influential ICFP Paper Award 2010[†]

> *"The techniques described in the paper have spawned a significant body of follow-on work in test case generation. They have also been adapted to other languages..."*

[†] See www.sigplan.org/Awards/ICFP/.

Testing with QuickCheck

An open source library

QuickCheck on Hackage.[†]

[†] http://hackage.haskell.org/package/QuickCheck.

Testing with QuickCheck

An open source library

QuickCheck on Hackage.[†]

Commercialisation

QuviQ (www.quviq.com/).

[†] http://hackage.haskell.org/package/QuickCheck.

Testing with QuickCheck

Adaptations

QuickCheck has been ported to various languages (Wikipedia 2025-07-28).

C	C#	C++	Chicken	Clojure
Common Lisp	Coq	D	Elm	Elixir
Erlang	F#	Factor	Go	Io
Java	JavaScript	Julia	Logtalk	Lua
Mathematica	Objective-C	OCaml	Perl	Prolog
PHP	Pony	Python	R	Racket
Ruby	Rust	Scala	Scheme	Smalltalk
Standard ML	Swift	TypeScript	VB.NET	Vhiley

Testing with QuickCheck

False positive

The program works properly but the test pointed out a fail:

- There is a bug elsewhere.
- There is an error in the specification.

Testing with QuickCheck

False positive

The program works properly but the test pointed out a fail:

- There is a bug elsewhere.
- There is an error in the specification.

False negative

There is a bug in the program but the test passed.

Recall Dijkstra's 1969 famous quote:

> *"Program testing can be used to show the presence of bugs, but never to show their absence!" (Dijkstra 1970)*

Testing with QuickCheck

Example

See demo.

References

E. W. Dijkstra (1970). Structured Programming. In: Software Engineering Techniques (NATO Software Engineering Conference 1969). Ed. by J. N. Buxton and B. Randell, pp. 84–88 (cit. on pp. 7, 8).