QuviQ

Using QuickCheck for Testing C Code

Ulf Norell

Erlang Factory 2009, London

Scratch

- Motivation

- Goal
  - fine grained testing for C programs
  - as easy as testing Erlang programs
- Demo
  - simple but convincing example (Stack)
  - structs as records
  - typed pointers
  - primitives: alloc, create_array, deref, read_array
  - catching segmentation faults
- How does it work?
  - parsing the C program, generating wrappers (both C and Erlang)
  - running C program in a separate OS process (crashing doesn’t affect the Erlang node)

Goals

- Fine grained testing of C code
- Calling C functions as if they were Erlang functions
- Minimal manual work

- Work in progress…

Demo

Overview

- Parses your C program, generating
  - Erlang wrappers for the C functions
  - Primitive functions for memory management (alloc/deref) for user defined types
  - A C program that is running as a separate process
- C structs modelled as Erlang records
- Pointers are tagged with their type
- C exceptions are translated to Erlang exceptions
  - e.g. segmentation faults

Future work

- Try more interesting examples
- Generate Erlang macros from #defines
- C++