Bhavin Shah

bhavin@mail.rit.edu | +1 585-629-3058 | Linkedin://scanbhavin | Github://scanbns

EDUCATION

Master of Science in Computer Science, Rochester Institute of Technology, NY **Expected Dec 2018**

Courses: Compiler Construction, Algorithms, Programming Languages Theory, Graph Theory, Computer Networks, Haskell

Bachelor of Engineering in Information Technology, University of Mumbai, India **May 2012**

Coursera, Functional Programming Principles in Scala, EPFL

SKILLS

Languages - Java, Scala, Haskell, Python, Elixir, C++, JavaScript, REST, HTML, CSS

Databases - SQL, Neo4j, Cassandra, Postgres

Frameworks - Phoenix, Spring, Struts, Eclipse RCP

Tools - Maven, SBT, Ant, Git, Coq, Linux, Emacs, Vim, Windows, Jenkins, JIRA, YourKit Java Profiler

EXPERIENCE

Software Engineer Intern, Terakeet, Syracuse NY May 2017 – Aug 2017

- Developed an analytic and tracking framework that listens for various user and account actions for a social network that allows brands to collaborate with influencers. The data collected is then posted to Datadog.
- Implemented a REST API to link and access users social account using developer API from Facebook, Twitter, LinkedIn, Instagram, and Pinterest.

Software Engineer, Itiviti Feb 2014 – Jul 2016

- Built an Order Management System to trade on Istanbul Bourse. Designed pluggable modules that would validate an order, check risk limit (gross exposure/fat finger) and split large orders into multiple smaller parts. On the return path, the split order parts are reconciled to a single order. Incorporated performance improvements to bring down the round-trip latency below 0.7ms per order.
- Implemented a graphical XML editor in eclipse RCP to create and edit user interfaces for algo strategies in FIXatdl format.
- Implemented market data handling system to manage user subscriptions for stocks (market data) and publish tick by tick updates.

Software Consultant, Dark Horse IT Consulting Dec 2012 – Jan 2014

- Developed solutions for insurance claims pay-out application using JavaScript and Oracle database.
- Implemented split screen functionality in that increased data entry efficiency by 30 percent.

Grader - Compiler Construction, Rochester Institute of Technology Jan 2018 – present

PROJECTS

Java Compiler - Constructed a compiler for e-mini Java language through various logical phases such as Lexical analysis, Syntax analysis, Name analysis, Type checking, Byte code generation and Optimization. Implemented a recursive-decent-parser and optimizations like constant folding to reduce size of the generated executable file.

Auto-Program-Verifier - Implemented a program verification engine to verify a simple imperative language. It generates control flow graph and a set of Horn clauses for the program and verifies if the program has a bug or not. The program is verified by consulting Eldarica Horn solver, a model checker for Horn clauses.

Sokoban - Implemented Sokoban game in Haskell involving moving crates through a complex maze into storage locations. (bit.ly/playsokoban)

SAT Solver - Implemented an efficient Boolean Satisfiability Solver using DPLL algorithm. It takes in a Boolean formula in CNF format and determines if there exists an interpretation that satisfies the given formula.

Simple-Type-Checker - Implemented type checking and type inference on a simple λ-calculus language. The program infers type for a given expression i.e. it either finds a type error in an expression, or, it infers a fully determined type.

QJump - Implemented a custom queuing discipline in Linux Traffic Control Module to reduce network interference and provide guaranteed latency in data center networks. It works by rate-limiting the input into the network, so that long queues cannot build up; and prioritizing traffic in the network, so that different applications can use rate-limits that suit them best.

Routing-Info-Protocol - Implemented active RIPv2 (distance-vector routing protocol) that recovers from failed links/routers and avoids count to count to infinity problem using split-horizon routing with poison reverse.