Finite State Machines + Deep Learning + Minsky Machines in the Context of Information Processing Using : Scala + JVMs + LLVM in Heterogeneous Computing Environments - A Suggestion.

Nirmal - Informatics R&D - Current Member - antE Inst UTD Dallas TX USA.
email id : hmfg2014@gmail.com

Abstract :

Testing of Finite State Machines [FSM] + Deep Learning [DL] + Minsky Machines [MM] w.r.t Scala Programming Language/JVMs/Metascala VM involving : Smart Devices [SD] + IoT + High Performance Computing -> A Simple Technical Communication on Using : HOL-Isabelle System + JikesRVM-Reseearch Virtual Machine + LLVM + Scala + LMS-Scala + Applied Mathematics + Multi-Media IoT Informatics & BIG DATA.NOT all the details are covered in this short technical communication.We hope our readers will dig deep into the topics/algorithms mentioned here.

index words / keywords : FSM ; DL ; MM ; Scala ; RVM ; HPC ; BIG DATA.

[I] Main Idea + Inspiration + Introduction :

"This entry provides executable formalization of complete test generation algorithms for nite state machines. It covers testing for the language-equivalence and reduction conformance relations, supporting the former via the W, Wp,HSI, H, SPY and SPYH-methods, and the latter via adaptive state counting. The test strategies are implemented using generic frameworks, allowing for reuse of shared components between related strategies.This work is described in the author's doctoral thesis."

[***Source -> https://www.isa-afp.org/entries/FSM_Tests.html***]

"An Insight into HOL-Isabelle/Coq Theorem Provers based Design of Algorithms Using [Minsky Machines+Scala NLP/Scala/Akka/JikesRVM-Research Virtual Machine/JVM/LLVM] in the Context of Electronic Health Record [EHR] Software R&D - A Simple Suggestion on Using [NLP + IoT + HPC]."

[***Source-> https://www.vixra.org/pdf/1909.0490v1.pdf***]

"FORMALIZING IMAGE PROCESSING IN HIGHER ORDER LOGIC(HOL) BY UNDERSTANDING AND USING XML-HOL-SCALA-JVM SOFTWARE FRAMEWORK TOWARDS PROCESSING OF CRYO-EM/TEM/SEM IMAGES BASED ON LEVY PROCESSES - A NOVEL SUGGESTION IN THE CONTEXT OF INSTRUMENTATION & HIGH PERFORMANCE COMPUTING ENVIRONMENT."

[***Source -> https://vixra.org/pdf/1709.0412v1.pdf***]

[II] R&D Informatics Framework to TEST DL & FSMs & MMs :

Please generate your own Frameworks.

[III] Additional Information on Mathematics + Software Used :

[a] https://vixra.org/pdf/2001.0015v1.pdf

[b] https://www.isa-afp.org/entries/Deep_Learning.html

[c] *https://www.isa-afp.org/entries/FSM_Tests.html*

[d] https://isabelle.in.tum.de/

[e] https://www.isa-afp.org/entries/Minsky_Machines.html

[f] http://pi4.informatik.uni-mannheim.de/~kopf/publications/publications.html

[g] https://github.com/tejdnk-2019-ShortNotes -> Plenty of Examples.

[IV] Simple Review + Discussion/s on JVM Languages + Other Related Information :

"Scala combines object-oriented and functional programming in one concise, high-level language. Scala's static types help avoid bugs in complex applications, and its JVM and JavaScript run times let you build high-performance systems with easy access to huge ecosystems of libraries."

[***Source -> https://www.scala-lang.org/***]

"Jikes RVM (Research Virtual Machine) provides a flexible open test-bed to prototype virtual machine technologies and experiment with a large variety of design alternatives. The system is licensed under an OSI approved license. Jikes RVM runs on many platforms and advances the state-of-the-art of virtual machine technologies for dynamic compilation, adaptive optimization, garbage collection, thread scheduling, and synchronization. A distinguishing characteristic of Jikes RVM is that it is implemented in the Java™ programming language and is self-hosted i.e., its Java code runs on itself without requiring a second virtual machine. Most other virtual machines for the Java platform are written in native code (typically, C or C++). A Java implementation provides ease of portability, and a seamless integration of virtual machine and application resources such as objects, threads, and operating-system interfaces."

[***Source -> https://www.jikesrvm.org/***]

"Hera-JVM: abstracting processor heterogeneity behind a virtual machine McIlroy, R. and Sventek, J. (2009) Hera-JVM: abstracting processor heterogeneity behind a virtual machine. In: 12th Workshop on Hot Topics in Operating Systems, Monte Verità, Switzerland, 18 - 20 May 2009."

[Source -> https://eprints.gla.ac.uk/6231/]

IoT :

Portable devices that feature Internet connectivity are very common these days. Since Java was created with the idea of running on embedded devices from the beginning, JVM is, yet again, at an advantage here.

For memory-constrained systems, Oracle offers the Java ME Embedded platform. It is meant for commercial IoT devices that do not require a standard graphical or console-based user interface.

For devices that can spare more memory, the Java SE Embedded edition is available. The Java SE Embedded version is very close to the Java SE discussed in this book. When running a full Linux environment, it can be used to provide desktop GUIs for full user interaction.

Both Java ME Embedded and Java SE Embedded platforms can access the general-purpose ...”

[***Source -> https://www.oreilly.com/library/view/introduction-to-jvm/9781787127944/5ac1c37c-da73-40ee-95b5-5aa88342b239.xhtml***]

Scala-IOT : Introduction to Internet Of Things :

*"*As wikipedia says, The **internet of things** (**IoT**) is the network of physical devices, vehicles, buildings and other items—embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.”

[Source -> https://blog.knoldus.com/iot-introduction-to-internet-of-things/]

Creating Secure IoT device identities :

“Device Identities are More Complex Than a Simple Certificate The Internet of Things (IoT) is growing exponentially, but security for IoT projects and deployments remains an obstacle for many organizations. One fundamental IoT security component is making sure devices and services have trusted identities that can interact within secure ecosystems.”

[***Source -> https://www.intertrust.com/resources/creating-secure-iot-device-identities/***]

DSLs - A Powerful Scala Feature :

“Domain Specific Languages (DSL) written in Scala have become powerful tool in the hands of Scala programmers. In the original language design great care was taken to ensure that the syntax would allow programmers to create natural looking DSLs. Spend a moment to enjoy Michael Fogus's humour and the clever use he makes of implicits to create "Baysick". It looks like BASIC feels like BASIC but it's a Scala DSL. Then you may like to follow some other links to see DSL examples for Financial Asset Management, Apache Camel Services and OSGi.”

[Source -> https://www.scala-lang.org/old/node/1403]

"**Lightweight Modular Staging (LMS)** is a run-time code generation approach. The framework provides a library of core components for building high performance code generators and embedded compilers in Scala.”

[Source -> https://scala-lms.github.io/]

LLVM Compiler Tool Kit :

"The LLVM Project is a collection of modular and reusable compiler and toolchain technologies. Despite its name, LLVM has little to do with traditional virtual machines. The name "LLVM" itself is not an acronym; it is the full name of the project.LLVM began as a research project at the University of Illinois, with the goal of providing a modern, SSA-based compilation strategy capable of supporting both static and dynamic compilation of arbitrary programming languages. Since then, LLVM has grown to be an umbrella project consisting of a number of subprojects, many of which are being used in production by a wide variety of commercial and open source projects as well as being widely used in academic research. Code in the LLVM project is licensed under the "Apache 2.0 License with LLVM exceptions" “

[Source -> https://llvm.org/]

[V] Acknowledgment/s :

Sincere Thanks to all WHO made this happen in my LIFE.Inspire others always.Non-Profit R&D Only.

[VI] Conclusions/ + Future Perspectives :

One of the pioneering R&D Efforts in this domain of AI based Complex Embedded Systems + Advanced Information Processing.

In the Near Future - We might also take a look at some of the features of HOL Theorem Prover on the following lines involving our Novel Algorithms mentioned above :

"
HOL: various library improvements (***HOL-Library, HOL-Combinatorics,HOL-Analysis, HOL-Statespace***)

ML: uniform treatment of external processes via Isabelle/Scala.

System: support for Isabelle/Scala modules defined in user-space.

System: improved document preparation using Isabelle/Scala.

System: update to Poly/ML 5.9 with improved support for ARM64 on Linux."

[VII] References :

1. T. S. Chow. Testing software design modeled by finite-state machines. IEEE Transactions on Software Engineering, 4(3):178–187, 1978.

2. G. Luo, G. von Bochmann, and A. Petrenko. Test selection based on communicating nondeterministic finite-state machines using a generalized wp-method. IEEE Trans. Software Eng., 20(2):149–162, 1994.

3. G. Luo, A. Petrenko, and G. von Bochmann. Selecting test sequences for partially-specified nondeterministic finite state machines. In T. Mizuno, T. Higashino, and N. Shiratori, editors, Protocol Test Systems: 7th workshop 7th IFIP WG 6.1 international workshop on protocol text systems, pages 95–110. Springer US, Boston, MA, 1995.

4. R. Dorofeeva, K. El-Fakih, and N. Yevtushenko. An improved conformance testing method. In F. Wang, editor, Formal Techniques for Networked and Distributed Systems - FORTE 2005, 25th IFIP WG 6.1 International Conference, Taipei, Taiwan, October 2-5, 2005, Proceedings, volume 3731 of Lecture Notes in Computer Science, pages 204–218. Springer, 2005.

5. A. da Silva Simão, A. Petrenko, and N. Yevtushenko. On reducing test length for fsms with extra states. Software Testing, Verification and Reliability, 22(6):435–454, 2012.

6. M. Soucha and K. Bogdanov. Spyh-method: An improvement in testing of finite-state machines. In 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops, ICST Workshops, Västerås, Sweden, April 9-13, 2018, pages 194–203. IEEE Computer Society, 2018.

7. A. Petrenko and N. Yevtushenko. Adaptive testing of nondeterministic systems with FSM. In 15th International IEEE Symposium on High-Assurance Systems Engineering, HASE 2014, Miami Beach, FL, USA, January 9-11, 2014, pages 224–228. IEEE Computer Society, 2014.

[THE END]